



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Facultat d'Informàtica de Barcelona



# Visualització avançada de models mèdics

*Josep Ciurana Herrera*

UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) - BARCELONATECH

FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)

GRAU EN ENGINYERIA INFORMÀTICA

ESPECIALITAT DE COMPUTACIÓ

TREBALL DE FI DE GRAU

**Dirigit per**

PERE PAU VÁZQUEZ ALCOCER  
Dpt. Ciències de la Computació

**Codirigit per**

EVA MONCLÚS LAHOYA  
Grup de recerca ViRVIG

2 de juliol de 2020



## Resum

L'increment de la capacitat computacional de les targetes gràfiques i l'aparició de cascs de realitat virtual a preus relativament assequibles els darrers anys, ha permès el desenvolupament i la expansió de la exploració interactiva de models volumètrics en un entorn de realitat virtual immersiva.

Aquest recurs, que actualment s'usa en diversos àmbits com la medicina o el disseny industrial, presenta certs problemes. Un dels més importants és com passar dels valors d'entrada del model a la representació gràfica que veiem i com podem modificar aquesta representació interactivament. Per fer aquesta traducció s'utilitza el que es coneix com a funció de transferència, que en la seva forma més bàsica relaciona el valor d'entrada a una opacitat i color. Aquest tipus de funcions de transferència, funcionen prou bé si es defineixen abans d'explorar el model però la seva modificació interactiva durant l'exploració és bastant incòmoda.

L'objectiu d'aquest projecte consisteix en comprovar si l'ús d'una metàfora d'interacció usant *style transfer function*, un tipus de funcions de transferència que permeten una major abstracció de les dades i presenta un acabat similar a les imatges il·lustratives dels llibres mèdics, resulta més còmode i entenedor per l'usuari que una metàfora d'interacció usant una funció de transferència unidimensional.

## Resumen

El incremento de la capacidad computacional de las tarjetas gráficas y la aparición de cascos de realidad virtual a precios relativamente asequibles los últimos años, ha permitido el desarrollo y la expansión de la exploración interactiva de modelos volumétricos en un entorno de realidad virtual inmersiva.

Este recurso, que actualmente se usa en diversos ámbitos como la medicina o el diseño industrial, presenta ciertos problemas. Uno de los más importantes es como pasar de los valores de entrada del modelo a la representación gráfica que vemos y cómo podemos modificar esta representación interactivamente. Para hacer esta traducción se utiliza lo que se conoce como función de transferencia, que en su forma más básica relaciona el valor de entrada a una opacidad y color. Este tipo de funciones de transferencia, funcionan suficientemente bien si se definen antes de explorar el modelo pero su modificación interactiva durante la exploración es bastante incomoda.

El objetivo de este proyecto consiste en comprobar si el uso de una metáfora de interacción usando *style transfer function*, un tipo de funciones de transferencia que permiten una mayor abstracción de los datos y presentan un acabado similar a las imágenes ilustrativas de los libros médicos, resulta más cómoda y entendedora para el usuario que la metáfora de interacción usando una función de transferencia unidimensional.



## Abstract

The increase in the computing capacity of graphics cards and the emergence of virtual reality headsets at relatively affordable prices in recent years has allowed the development and expansion of interactive exploration of volumetric models in an immersive virtual reality environment.

Volume models, which are currently used in various fields such as medicine or industrial design, has certain problems. One of the most important is how to move from the input values of the model to the graphical representation that we see and how we can modify this representation interactively. For establishing this correspondence, what is known as a transfer function is used, which in its most basic way relates the input value to an opacity and a color. These types of transfer functions work well if they are defined before the user explores the model in real-time, but their interactive modification during exploration is quite cumbersome.

The purpose of this project is to test if the use of an interaction metaphor using *style transfer function*, a type of transfer function that allows for greater abstraction of data and has a similar finishing than the illustrative images of medical books, is more comfortable and user-friendly than an interaction metaphor using a one-dimensional transfer function.



# Sumari

<b>1</b>	<b>Introducció i contextualització</b>	<b>1</b>
1.1	Definició de conceptes fonamentals . . . . .	1
1.1.1	Renderitzat de volums . . . . .	2
1.1.2	Realitat virtual . . . . .	3
1.2	Descripció del problema . . . . .	4
1.3	Actors implicats . . . . .	5
<b>2</b>	<b>Abast del projecte</b>	<b>7</b>
2.1	Objectius . . . . .	8
2.1.1	Metàfora d'interacció amb <i>style transfer function</i> . . . . .	8
2.1.2	Metàfora d'interacció amb funció de transferència unidimensional . . . . .	8
2.1.3	Test de validació . . . . .	8
2.2	Obstacles i riscos . . . . .	8
2.3	Metodologia i rigor . . . . .	9
2.3.1	Mètode àgil . . . . .	9
2.3.2	Eines . . . . .	9
2.3.3	Validació del projecte . . . . .	10
<b>3</b>	<b>Desenvolupament de les metàfores</b>	<b>11</b>
3.1	Introducció al projecte original . . . . .	11
3.2	Metàfora 1D_TF . . . . .	13
3.2.1	Funcions de transferència unidimensionals . . . . .	13
3.2.2	Disseny i implementació . . . . .	14
3.2.3	Interacció . . . . .	15
3.3	Metàfora Style_TF . . . . .	17
3.3.1	<i>Style transfer functions</i> . . . . .	17
3.3.2	Disseny i implementació . . . . .	19
3.3.3	Interacció . . . . .	23
3.4	Anàlisi de rendiment . . . . .	24
3.5	Resultat final . . . . .	24
<b>4</b>	<b>Test de validació</b>	<b>29</b>
4.1	Disseny del experiment . . . . .	29
4.1.1	Seguiment del usuari . . . . .	29
4.1.2	Qüestionari . . . . .	30
<b>5</b>	<b>Gestió del projecte</b>	<b>31</b>
5.1	Planificació temporal . . . . .	31
5.1.1	Definició de les tasques . . . . .	31
5.1.2	Representació gràfica de la planificació . . . . .	34
5.1.3	Gestió del risc . . . . .	35
5.2	Gestió econòmica . . . . .	35
5.2.1	Costos de personal per activitat . . . . .	35
5.2.2	Costos genèrics . . . . .	37
5.2.3	Costos generals del projecte . . . . .	38
5.2.4	Altres costos . . . . .	38
5.2.5	Costos totals del projecte . . . . .	39

5.2.6	Control de gestió . . . . .	39
5.3	Sostenibilitat i compromís social . . . . .	39
5.3.1	Autoavaluació . . . . .	39
5.3.2	Dimensió ambiental . . . . .	39
5.3.3	Dimensió econòmica . . . . .	40
5.3.4	Dimensió social . . . . .	40
<b>6</b>	<b>Conclusió</b>	<b>41</b>
	<b>Bibliografia</b>	<b>42</b>

# Índex de figures

1.1	Representació tridimensional d'un crani humà aplicant renderitzat de volums . . . . .	1
1.2	Il·lustració d'un sistema de tomografia computeritzada . . . . .	2
1.3	<i>Pipeline</i> gràfic del renderitzat de volums . . . . .	2
1.4	Dibuix il·lustratiu del funcionament de la tècnica de <i>ray casting</i> . . . . .	2
1.5	Exemple de segmentació i anàlisi volumètric de les cambres cardíques . . . . .	3
1.6	Representació d'ús de les dos aproximacions a la VR . . . . .	4
1.7	Cirurgians explorant la zona afectada amb un sistema de VR abans d'una operació . . . . .	5
2.1	Exemple d'ús de les <i>style transfer functions</i> . . . . .	7
2.2	Comandament de les HTC Vive amb descripció dels botons . . . . .	9
3.1	<i>Pipeline</i> simplificat del projecte original . . . . .	11
3.2	Dibuixos il·lustratius de la tècnica de <i>ray casting</i> amb models volumètrics . . . . .	12
3.3	Comparativa amb el mateix model volumètric visualitzant-se amb i sense segmentació . . . . .	13
3.4	Gràfiques que relacionen intensitat i opacitat amb un rang definit . . . . .	14
3.5	<i>Pipeline</i> simplificat de la metàfora 1D_TF . . . . .	15
3.6	Interfície amb nom <i>UI 1D_TF</i> creada a Unity per editar la TF interactivament . . . . .	15
3.7	Diagrama d'interacció de la metàfora 1D_TF . . . . .	16
3.8	Passos a realitzar per modificar la TF amb la metàfora 1D_TF . . . . .	16
3.9	Diferents visualitzacions del mateix model definides usant la interacció proposada . . . . .	17
3.10	Procés d'obtenció del mapa de les <i>lit spheres</i> . . . . .	17
3.11	Resultat d'aplicar un estil amb contorn a un model volumètric . . . . .	18
3.12	<i>Pipeline</i> per obtenir el color de la <i>lit sphere</i> . . . . .	18
3.13	<i>Pipeline</i> simplificat de la primera fase de la metàfora Style_TF . . . . .	19
3.14	<i>Pipeline</i> simplificat de la segona fase de la metàfora Style_TF . . . . .	20
3.15	Diagrama UML de la classe <i>Style_VolumeRaytraceEffect</i> . . . . .	20
3.16	Interfície amb nom <i>UI Style_TF</i> creada a Unity per editar la STF interactivament . . . . .	21
3.17	Demostració de la capacitat de ressaltar segments assenyalats amb el punter . . . . .	22
3.18	Interfícies de les dues aproximacions diferents a la interacció amb el model segmentat . . . . .	22
3.19	Diagrama UML de la classe <i>StyleMetaphorInteractionController</i> . . . . .	25
3.20	Diagrama d'interacció de la metàfora Style_TF amb models segmentats . . . . .	26
3.21	Passos a realitzar per modificar l'estil i la opacitat d'un segment amb la 1a aproximació . . . . .	26
3.22	Passos a realitzar per modificar l'estil i la opacitat d'un segment amb la 2a aproximació . . . . .	27
3.23	Visualització del mateix model volumètric usant les tècniques desenvolupades . . . . .	27
3.24	Diferents visualitzacions del mateix model usant la metàfora 1D_TF . . . . .	28
3.25	Diferents visualitzacions del mateix model usant la metàfora Style_TF (sense segmentació) . . . . .	28
3.26	Diferents visualitzacions del mateix model usant la metàfora Style_TF (amb segmentació) . . . . .	28
5.1	Gantt il·lustratiu de la planificació temporal inicial del projecte . . . . .	34
5.2	Gantt il·lustratiu de la planificació temporal final del projecte . . . . .	34

# Índex de taules

5.1	Llistat de les tasques del TFG amb l'estimació temporal . . . . .	32
5.2	Predicció de costos de personal en funció del perfil . . . . .	36
5.3	Retribucions per cada un dels perfils que participa en el projecte . . . . .	37
5.4	Cost dels recursos <i>hardware</i> . . . . .	37
5.5	Cost del consum elèctric del <i>hardware</i> . . . . .	38
5.6	Predicció dels costos generals . . . . .	38
5.7	Costos generats per imprevistos . . . . .	38
5.8	Predicció dels costos totals del projecte . . . . .	39

# Índex d'algorismes

1	Volume rendering with ray casting for 1D TF . . . . .	12
2	Segmented volume rendering with ray casting for 1D TF . . . . .	13
3	<i>Style transfer function</i> lookup . . . . .	19
4	Volume rendering with ray casting for STF . . . . .	21
5	Get segmentation identifier . . . . .	23
6	Segmented volume rendering with ray casting for STF . . . . .	24





# Capítol 1

## Introducció i contextualització

La revolució tecnològica de les últimes dècades ha condicionat l'evolució de la majoria d'àmbits humans [1], com ara les comunicacions, la cultura, les relacions socials o la medicina. En aquest últim cas, els avanços informàtics, tan de *hardware* com de *software*, i l'ús d'innovadores tecnologies de captació de dades han impulsat la creació de noves àrees mèdiques, com la que emmarca aquest Treball de Fi de Grau (TFG), la imatge mèdica [2]. Aquesta consisteix en la representació visual en dues o tres dimensions (2D, 3D) del cos humà, no només de l'exterior sinó també de l'interior, a partir de dades obtingudes mitjançant alguna tecnologia mèdica com la ressonància magnètica (MRI)<sup>1</sup>, els ultrasons (US) o la tomografia computeritzada (CT). En la Figura 1.1 es presenta un exemple il·lustratiu.

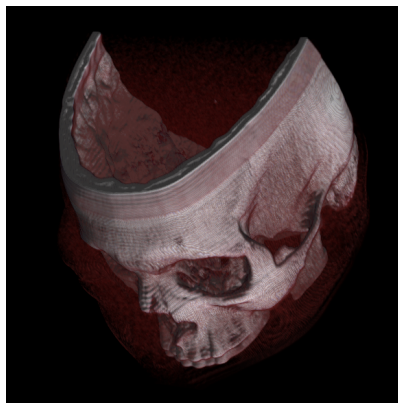


Figura 1.1: Representació tridimensional d'un crani humà aplicant una tècnica de renderitzat de volums al model volumètric obtingut amb CT. [3]

És en part gràcies a aquesta àrea que els sistemes de diagnosi i exploració del cos humà han experimentat un progrés significatiu. Tecnologies com la CT de la Figura 1.2 permeten explorar regions anatòmiques extenses en pocs segons, o crear imatges en moviment de zones clau, com el cor, amb diverses finalitats. A més, aquest tipus d'imatges 3D permeten estudiar i mesurar estructures d'interès de forma molt més precisa que d'altres eines, a diferència de les radiografies 2D, millorant així la planificació i les possibilitats d'èxit en intervencions.

Encara que la imatge mèdica pertany a un grup multidisciplinari, un dels seus principals fonaments és la informàtica i, en concret, camps que associem amb la computació com són els gràfics, la intel·ligència artificial o l'anàlisi d'eficiència. Aquest projecte treballa la representació de volums i com l'usuari pot modificar la manera en què es visualitzen a partir de la interacció.

### 1.1 Definició de conceptes fonamentals

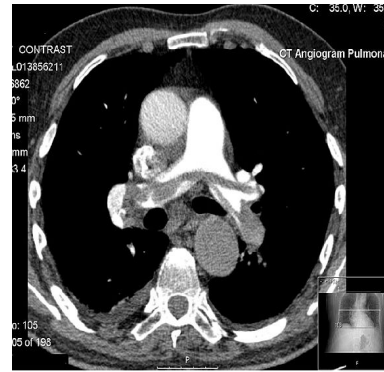
Un cop feta la introducció al tema que ens ocupa, procedim a definir els conceptes, tècniques i sistemes que considerem essencials per poder comprendre el projecte encara que no es disposin de coneixements previs específics de la matèria.

---

<sup>1</sup>Encara que els noms es presenten en català, s'ha decidit usar els acrònims anglesos ja que així apareixen en la majoria de documents tècnics.



(a) Aparell mèdic de captació de dades



(b) Visualització de les dades obtingudes

Figura 1.2: Il·lustració d'un sistema de tomografia computeritzada. [4] [5]

### 1.1.1 Renderitzat de volums

Renderitzat de volums [6] és un terme que agrupa totes les tècniques usades per representar un conjunt de dades discretament mostrejades en 3D (podeu veure el *pipeline* complet a la Figura 1.3). Aquestes dades reben el nom de models volumètrics i presenten una estructura diferent a la dels models geomètrics d'ordinador més comuns, formats per malles de triangles. La seva creació es pot interpretar com l'apilament d'imatges 2D, obtingudes mitjançant alguna tècnica com ara la CT, fins a formar un objecte 3D que acostuma a tenir forma de cub.

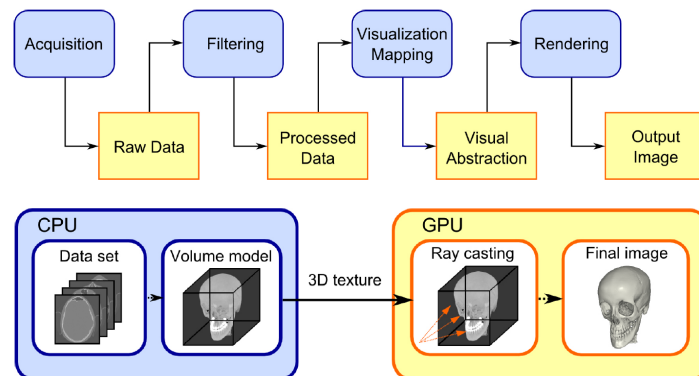
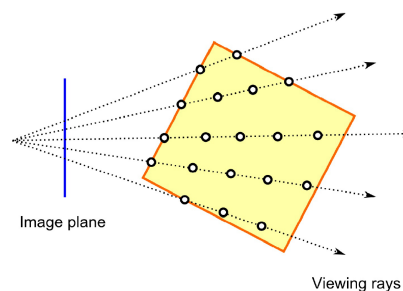


Figura 1.3: Pipeline gràfic del renderitzat de volums, des de l'adquisició del conjunt de dades fins a la imatge final. [7]

Les tècniques de renderitzat de volums es poden classificar en dos grups, indirecte i directe. En el nostre cas estem interessats en una tècnica pertanyent al segon grup, el *ray casting*. Aquesta consisteix en produir un raig per cada píxel de la finestra de visualització, el qual recorre l'escena i pren mostres del model volumètric en intervals regulars (veure Figura 1.4).

Figura 1.4: Dibuix il·lustratiu del funcionament de la tècnica de *ray casting*. [7]

Ara ja sabem com representar aquest tipus de models però, generalment, això és només una part del procés. El que es desitja és traduir els valors d'aquests models, els quals acostumen a ser escales de densitat (que es poden interpretar com a intensitats), per tal d'obtenir una representació més clara i, per exemple, ajudar al radiòleg a diagnosticar una malaltia més fàcilment que amb les dades en cru o ajudar a un cirurgià a planificar una intervenció quirúrgica. Aquí és on entren en acció les funcions de transferència (d'ara endavant, TF, de l'anglès *transfer function*), aquestes fan justament això, assignen a cada valor del model volumètric certes propietats per a la seva representació, com poden ser color i/o opacitat. Una bona analogia per entendre la tasca de les TF seria imaginar una imatge en blanc i negre (escala de grisos) que volem convertir a color, si a cada valor de gris li assignem un color, obtenim la següent funció matemàtica:

$$f: \mathbb{R} \rightarrow \mathbb{R}^3$$

$$x \mapsto f(x)$$

Aquesta seria la nostra funció de transferència, on per cada valor de gris de la imatge original obtenim un color (descriu usant el model de color RGB).

### Volums segmentats

En alguns casos, com per exemple en la identificació de tumors, es necessari recórrer a l'ús de models volumètrics segmentats o classificats (veure Figura 1.5). Tenir segmentació vol dir que cada posició del cub té associada un identificador que identifica l'estructura anatòmica més prevalent en aquell vòxel, el que permet la divisió del volum en segments on cada un presenta el mateix identificador. Algunes segmentacions també tenen en compte la frontera, zona on acaba un segment i en comença un altre, reservant uns identificadors concrets per aquestes zones. L'obtenció d'aquests models s'acostuma a fer amb tècniques semiautomàtiques, les quals impliquen certa participació del usuari en la segmentació o classificació (per exemple indicant al programa certes estructures d'interès), a partir dels models volumètrics basats en intensitat.

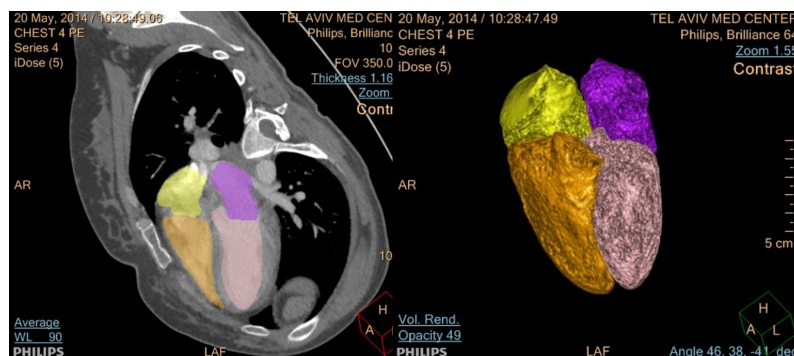


Figura 1.5: Exemple de segmentació i anàlisi volumètric de les cambres cardíques. [8]

Utilitzar volums segmentats també implica certes modificacions en el *pipeline* de visualització. En aquest cas, el paràmetre d'entrada són identificadors i no intensitats. És un canvi relativament petit però en alguns casos també cal tenir present altres elements, com la segmentació de la frontera.

Per exemple, podríem tenir un model volumètric que representa un sistema mecànic complex, format per diverses peces, on cada una té assignada un identificador diferent, el que permetria pintar-les de diferents colors per tal de poder distingir-les fàcilment en la visualització.

En comparació amb la TF definida prèviament, la qual és una classificació molt bàsica per intensitat, els processos de segmentació acostumen a tenir funcions classificadores molt més complexes on intervenen múltiples variables. Tornant al exemple de la Figura 1.5, es pot observar com la visualització obtinguda no seria possible utilitzant únicament la intensitat (és molt semblant en els quatre segments).

### 1.1.2 Realitat virtual

La realitat virtual (VR) [9], com el seu nom indica, és una experiència simulada, la qual pot ser similar al món real però no necessàriament. Els ordinadors actuals ens permeten crear entorns tridimensionals i programar totes les interaccions que l'usuari podrà realitzar. Això s'acaba de completar amb *hardware* auxiliar que permet la immersió i la interacció.

Els dispositius de VR es poden classificar de dues formes, segons si són immersius o semi-immersius, i també si són dispositius d'entrada o de sortida de dades. Els dispositius semi-immersius disposen d'una

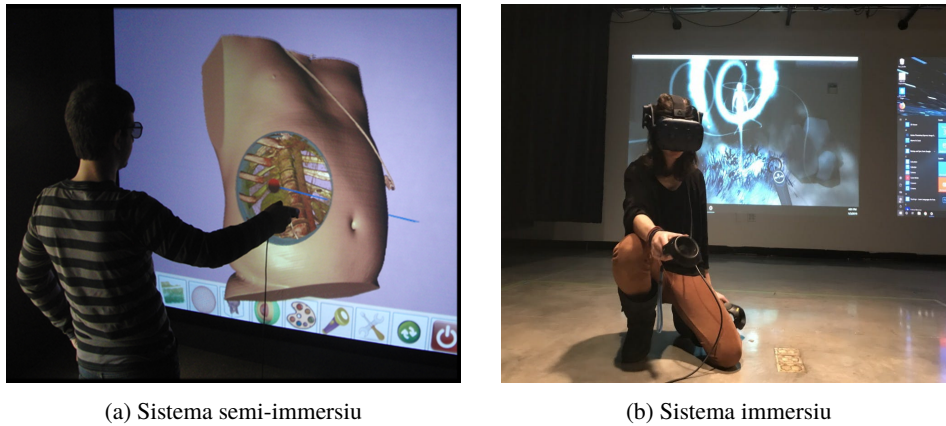


Figura 1.6: Representació d'ús de les dues aproximacions a la VR. [10] [11]

o més pantalles estèreo que simulen la presència d'objectes en un entorn des del punt de vista de l'usuari. En canvi, els sistemes immersius disposen d'una o més pantalles que es situen directament davant dels ulls de l'usuari, de tal manera que cada un veu una imatge lleugerament diferent, creant un efecte estereoscòpic que s'adequa al funcionament dels ulls humans. Els aparells anteriors són dispositius de sortida de dades, els d'entrada acostumen a ser comandaments i sensors que permeten la interacció amb l'entorn virtual.

La VR ha millorat molt en els últims anys. Com acostuma a ser freqüent [12], aquesta tecnologia es va desenvolupar en un principi per a usos militars, particularment en l'entrenament de soldats en simulacions. Afortunadament, la indústria del videojoc porta uns anys intentant establir una base d'usuaris que les utilitzi, el que ha portat a un abaratiment del preu dels dispositius d'VR immersiva. En un moviment que sembla l'evolució lògica després de la fallida de les pantalles 3D ara fa una dècada [13], empreses com Sony, Facebook o Valve tenen els seus propis dispositius de VR. A més, la indústria del videojoc també és la responsable del abaratiment i l'evolució de les targetes gràfiques destinades a ordinadors personals, el que ha beneficiat no només la VR sinó també altres sistemes computacionalment costosos, com el renderitzat de volums.

En els darrers anys han aparegut propostes de projectes que uneixen el renderitzat de volums amb la VR, els beneficis de poder explorar un objecte tridimensional en un entorn immersiu poden ser molt grans. Per exemple, en la cirurgia cerebral, una operació quirúrgica molt sofisticada, es fan servir sistemes de VR per planificar amb cura la millor forma d'accés a la part del cervell que s'ha d'intervenir. Cal tenir present que tant el renderitzat de volums com la VR són molt costosos computacionalment i, per aquesta raó, el seu ús conjunt implica tenir molta cura del rendiment del projecte.

## 1.2 Descripció del problema

En el marc de la inspecció de models volumètrics en un entorn de VR immersiu, en ser una tècnica novella, encara hi ha molts fronts de recerca oberts i poques pràctiques estandarditzades. Actualment, un dels camps on trobem moltes investigacions és el de l'edició de les TF [14], el problema que volem resoldre pertany a un subgrup dins d'aquesta gran àrea de recerca, el qual relaciona les TF amb la VR immersiva i destaca per la poca recerca consolidada.

El problema prové del fet que les TF, tot i ser conceptualment fàcils d'entendre, resulten molt difícils de definir en models volumètrics reals per tal d'obtenir representacions satisfactòries. Per exemple, si tornem a l'analogia anterior de donar color a una imatge, potser volem pintar unes herbes de verd però al definir la TF observem que també pintem el cel, l'herba i el cel comparteixen el mateix gris. O amb un exemple més real, en la classificació de teixits humans a partir de les intensitats, trobem teixits diferents que es mapen en un mateix rang d'intensitats i, per tan, si volem diferenciar-los ens cal recórrer a alguna altra característica diferenciadora, com el gradient. A això se li sumen altres complicacions, com ara que els responsables de dissenyar els programes no acostumen a representar el públic objectiu a qui van destinats, generalment són enginyers (amb l'ajut opcional de matemàtics) els que creen el *software* que després usen especialistes mèdics. Això produeix un xoc que pot arribar a afectar diversos àmbits, un dels més crítics és el de les TF, ja que si el programa permet modificar-les interactivament, l'edició amb les eines tradicionals resulta molt poc intuïtiva per algú que treballa en medicina.

Volem, per tant, trobar un sistema que ens permeti abstraure'ns de la representació típica de les TF

de manera que faciliti la claredat d'interpretació a un especialista mèdic i es mantingui un cert nivell d'interactivitat amb aquestes.

### 1.3 Actors implicats

Aquest projecte va dirigit als professionals mèdics, els quals podran usar-lo per inspeccionar la representació tridimensional del cos dels seus pacients per tal de poder seleccionar el tractament que millor s'adeqüi a les seves necessitats o preparar-se per a la seva intervenció quirúrgica (veure Figura 1.7).



Figura 1.7: Cirurgians explorant la zona afectada amb un sistema de VR abans d'una operació. [15]

A més, trobem altres col·lectius que també es poden beneficiar d'aquesta aplicació. Per exemple, estudiants de medicina podrien usar-lo per adquirir un ample ventall de coneixements relacionats amb el cos humà, les malalties i/o les lesions de manera dinàmica i entretinguda. També, es podria usar en museus o exposicions, on amb unes poques modificacions fins i tot podríem il·lustrar models volumètrics diferents dels mèdics com, per exemple, peces complexes en el camp del disseny assistit per computadora o models geo-tèrmics en el camp de la geologia.





## Capítol 2

# Abast del projecte

Per tal de determinar la millor solució al problema prèviament definit, es van consultar documents de recerca d'aquesta àrea. Primerament, es va usar un estat de l'art de la matèria [14] per tenir una idea general de la recerca en TF, la seva evolució en els últims anys i l'estat actual. Posteriorment, també es van buscar treballs que tractessin de resoldre problemes similars al nostre en el marc de la VR, però no es va trobar cap que abordés la problemàtica que ens plantejem aquí (en l'entorn de plataforma *desktop* les propostes són molt amplies).

Destaquem dos articles de recerca amb solucions que podrien ajudar-nos a solucionar el nostre problema, encara que no van dirigides a un entorn VR. El primer [16], planteja el mateix problema que el nostre i proposa una solució pensada per a la edició simplificada de TF en dispositius tàctils. El segon [17], introdueix una tècnica anomenada *style transfer functions* per representar TF usant textures generades a priori (veure Figura 2.1).

S'ha decidit adaptar la tècnica proposada en el segon article per un entorn de VR immersiu, encara que l'enfoc de la recerca no es exclusivament mèdic, creiem que el resultat s'adequa a una possible solució del problema definit.

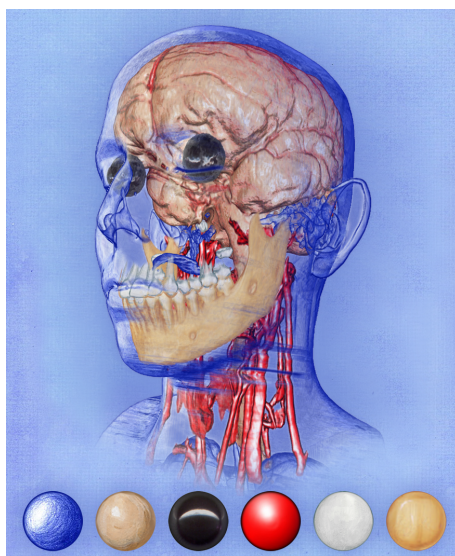


Figura 2.1: Exemple d'ús de les *style transfer functions*, les quals emulen l'estil il·lustratiu d'imatges de llibres, que utilitza 6 estils (veure part inferior de la imatge) amb un model volumètric segmentat. [17]

Cal tenir en compte que el nostre projecte ha partit d'una base donada, un projecte creat en Unity [18] i pensat per ser usat amb els cascs de VR HTC Vive [19], que és resultat d'anteriors TFG (en el nostre cas hem partit del TFG de Joan Fons Sánchez [20]), el qual s'ha ampliat. Això ha estat al mateix temps un ajut i un condicionant, partir d'aquest programa que ja visualitza volums en un entorn de VR immersiu ens ha permès arribar més lluny però també ens ha marcat com hem hagut d'implementar els nous elements (s'han seguit els esquemes de classes i el flux de dades definits en el TFG anterior).

## 2.1 Objectius

L'objectiu d'aquest TFG consisteix en provar la següent hipòtesi: l'ús d'una metàfora d'interacció<sup>1</sup> per la definició i l'edició de les TF en un entorn de VR immersiu basada en les *style transfer functions* és significativament més usable i el resultat és més satisfactori per a un especialista mèdic, com ara un radiòleg, que una adaptació literal de les eines de definició i edició de TF amb interfícies 2D (GUI clàssica d'una aplicació de *desktop*) per un entorn de VR immersiu. D'aquesta manera si la hipòtesi resulta correcta hauréu ajudat a alleugerar el problema que motiva aquest treball.

En el plantejament inicial, es pretenia desenvolupar un prototip i realitzar un test d'usuari per validar la hipòtesis plantejada però tenint en compte la crisi de la COVID-19, l'execució del test d'usuari amb un grup de persones externes al projecte no s'ha pogut dur a terme (això no vol dir que no s'hagi provat, jo mateix l'he utilitzat durant tot el desenvolupament). Finalment, s'ha desenvolupat el prototip (l'objectiu d'aquest projecte és la recerca, no un producte d'ús final) a partir del projecte actual de representació de models volumètrics en un entorn de VR immersiu, proporcionat pel grup de recerca. S'ha vetllat per un disseny i implementació que respectin el paradigma de disseny modular per tal de facilitar la seva interpretació en aplicacions futures. Aquest prototip està format per la metàfora d'interacció amb *style transfer function*, la metàfora d'interacció amb funció de transferència unidimensional i les eines per una validació adequada.

### 2.1.1 Metàfora d'interacció amb *style transfer function*

La metàfora d'interacció amb *style transfer function*, a partir d'ara anomenada metàfora Style\_TF, és l'objectiu principal del TFG, el qual ha consistit en la implementació de les *style transfer functions* en el projecte donat, adaptant el disseny de manera que facilitin el seu ús en VR. A més, un cop el sistema ha funcionat correctament, s'han ampliat les funcionalitats més enllà del que descriu l'article original, permetent l'edició interactiva d'aquestes funcions.

### 2.1.2 Metàfora d'interacció amb funció de transferència unidimensional

Per veure si efectivament la metàfora Style\_TF és més eficient i millora la usabilitat respecte el problema definit, també s'ha implementat una interfície d'edició i de modificació de TF en 2D tradicional (veure secció 1.1.1), adaptant-la a un entorn VR immersiu. Aquest sub-objectiu rep el nom de metàfora d'interacció amb funció de transferència unidimensional, abreviat com metàfora 1D\_TF.

### 2.1.3 Test de validació

Finalment, per tal de poder comparar, s'han afegit al prototip sistemes d'enregistrament de les accions que haurien d'haver realitzat els usuaris que anaven a avaluar el prototip durant el test de validació que s'hauria d'haver realitzat un cop implementades les dues metàfores. Com ja s'ha dit, l'execució del test no s'ha pogut realitzar degut a la pandèmia de la COVID-19.

Si s'hagués pogut realitzar l'avaluació de la tècnica implementada podríem haver conclòs si aquesta proposta millora l'experiència virtual del usuari. Les nostres conclusions podrien haver donat lloc a la producció d'un article científic. El risc de fallida és també una dada interessant des del punt de vista científic i, per tant, creiem que si finalment s'hagués pogut comprovar que la nostra hipòtesis és errònia aquest mateix fet hauria tingut valor per si mateix en el marc de la relació de les TF i la VR. Per desgràcia, sense l'execució del test no s'han tingut les dades necessàries per poder desenvolupar aquesta part.

## 2.2 Obstacles i riscos

Com ja s'ha comentat en el capítol anterior, un dels problemes amb que ens podíem trobar, i ens hem trobat, a l'hora de conceptualitzar és la poca informació que hi ha disponible actualment referent al ús conjunt de TF i VR. La metodologia triada i l'experiència del grup de recerca ha estat vital per minimitzar aquest fet.

Un altre punt limitant ha estat el rendiment, tant la tècnica de renderitzat de models volumètrics com l'ús de VR són característiques molt costoses des del punt de vista computacional. Pel renderitzat de volums s'usa la tècnica de *ray casting* que, com s'ha explicat, consisteix en llençar raigs des de la càmera al model volumètric, el nombre de raigs depèn de la resolució i això sumat al fet que en VR s'han de renderitzar les imatges de cada ull per separat (és doble la resolució horitzontal) i que s'han de mantenir un mínim de

<sup>1</sup>Procés que, en el nostre context, està format per la intenció, la gestualitat, la interpretació i el processament. Un exemple de metàfora força esclaridor és el de l'escriptori d'un ordinador (on s'usen icones per representar programes i el ratolí per seleccionar-los).



90 imatges per segon per no produir mareig evidència la rellevància d'aquest punt. Cada un d'aquests dos elements, la visualització de volums i la RV, poden arribar a saturar les GPUs actuals de *desktop* per separat però donat que en el nostre projecte s'han utilitzat conjuntament en la mateixa GPU, s'han hagut de fer implementacions eficients, descartat idees que resultaven molt costoses i acceptar que en algunes ocasions no es podien mantenir les 90 imatges per segon.

L'accés a les HTC Vive també ha sigut limitat, degut al seu relatiu alt preu (en cap cas comparable al preu dels dispositius que hi havia al mercat abans de l'arribada d'Oculus) s'han usat les de ViRVIG, a les quals s'ha tingut accés dos dies a la setmana. Aquest punt es va tornar crític en les últimes setmanes del projecte, ja que el confinament provocat per l'expansió del COVID-19 va impossibilitar tenir accés als cascs i, com a resultat, es va optar per endarrerir la presentació del TFG.

## 2.3 Metodologia i rigor

### 2.3.1 Mètode àgil

La metodologia de treball s'ha basat en un mètode àgil o *agile* [21] de cicles curts, aquesta metodologia de desenvolupament es basa en l'adaptabilitat de qualsevol canvi com a mitjà per augmentar les possibilitats d'èxit del projecte. En aquest sentit, cada setmana s'han realitzat dues reunions amb l'Eva Monclús Lahoya, codirectora i membre del grup de recerca ViRVIG, que han servit per fer el seguiment del treball realitzat, establir la feina a fer els propers dies i aclarir dubtes relacionats amb el programa proporcionat o la implementació d'alguna característica. La major part de la feina s'ha fet des de casa, s'han aprofitat els dies de reunió per testejar la feina realitzada usant l'entorn de VR al Centre de Realitat Virtual de la UPC, seu del grup d'investigació ViRVIG, aprofitant les HTC Vive de que disposen. També, de forma regular, s'han realitzat reunions de seguiment amb en Pere Pau Vázquez Alcocer, director del TFG. Amb aquesta metodologia, a més, s'ha evitat perdre molt temps en algunes situacions en les que s'ha avançat per una via errònia.

### 2.3.2 Eines

Es va crear una branca al repositori de GitLab [22] del projecte original de renderitzat de volums de ViRVIG des de la qual s'ha partit, s'ha utilitzat *git* com a sistema de control de versions. També es va crear un Trello [23] que ha servit per portar el seguiment i afegir els nous objectius amb periodicitat setmanal.

Com ja s'ha esmentat abans, la base del projecte s'ha realitzat amb el *game engine* Unity, concretament la versió 5.4.3f1 LTS. Pel codi font s'ha usat el llenguatge *C#* amb el suport dels IDEs Visual Studio i MonoDevelop, els quals s'han anat alternant en funció de les necessitats.

Pel que fa a la VR, l'ús de les HTC Vive implica recórrer a les llibreries SteamVR, versió 1.2.1, les quals presenten integració amb Unity.

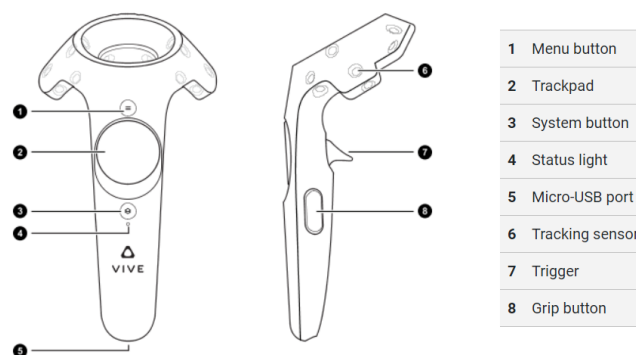


Figura 2.2: Comandament de les HTC Vive amb descripció dels botons. [19]

La memòria s'ha creat amb LaTeX, mitjançant el portal web Overleaf. Per gestionar la bibliografia s'ha usat la aplicació Mendeley.

### 2.3.3 Validació del projecte

Beneficiant-nos de la metodologia triada, una part de la validació del projecte s'ha dut a terme durant les reunions que han tingut lloc cada setmana. D'aquesta manera, s'ha planificat quines tasques fer cada setmana alhora que s'han validat les de l'anterior.

Per la validació de les dues metàfora inicialment es van plantejar dos sistemes. El primer, ha servit per veure com podem millorar les metàfores un cop ja tenim implementada una primera versió operativa. En aquest cas s'ha usat un petit grup de voluntaris, experts en la matèria, del propi grup de recerca per obtenir una opinió, a partir de la qual s'han valorat quins elements es podien millorar de cada metàfora. El segon sistema, el qual no s'ha pogut completar degut a la crisi de la COVID-19, era un test de validació que s'hauria d'haver realitzat amb les versions finals de cada metàfora i, en aquest cas, la intenció era comparar l'una amb l'altre. En aquesta ocasió, el grup d'usuaris que hauria d'haver avaluat el prototip era major i hauria inclòs personal mèdic per una aproximació més realista dels resultats. Es va arribar a desenvolupar una demostració específica per provar les metàfores 1D\_TF i Style\_TF, la qual enregistra les accions realitzades per permetre una valoració quantitativa. Posteriorment, s'hauria usat un qüestionari per obtenir dades qualitatives. Amb les dades obtingudes en aquest test s'hauria pogut realitzar una validació final del projecte desenvolupat i comprovar si la nostra hipòtesis és correcta.

## Capítol 3

# Desenvolupament de les metàfores

En aquest capítol s'explicarà en detall tot el procés de desenvolupament de les dues metàfores d'interacció definides en els objectius. Això engloba la definició teòrica dels dos tipus de funcions de transferència, el disseny i la implementació de classes i interfícies i les interaccions definides utilitzant la VR. Abans però, i per tal de poder comprendre millor el que s'ha desenvolupat, s'explicarà el funcionament bàsic del projecte proporcionat.

### 3.1 Introducció al projecte original

El nostre projecte parteix del resultat d'un altre, l'objectiu d'aquesta secció no és donar una visió detallada del projecte proporcionat (per això ja existeix la respectiva memòria) sinó permetre un enteniment global del seu funcionament. Per aconseguir-ho, la Figura 3.1 mostra una versió sintetitzada del *pipeline* de l'aplicació proporcionada (una versió completa i més general es pot veure en la Figura 1.3). Com es pot observar, l'aplicació carrega un conjunt d'arxius a la CPU, entre aquests trobem els que contenen el model i la TF. Aquesta aplicació, que utilitza TF unidimensional (veure definició en secció 3.2.1.), només permet la modificació de la TF durant l'execució usant un complement desenvolupat per Unity, és a dir, l'usuari que està explorant el model en l'entorn de VR no pot modificar-la, ha de ser un altre usuari des del ordinador. Amb les dades llegides dels arxius la CPU crea un conjunt de textures que passen als *shaders*<sup>1</sup> de la GPU com a *uniforms*<sup>2</sup>.

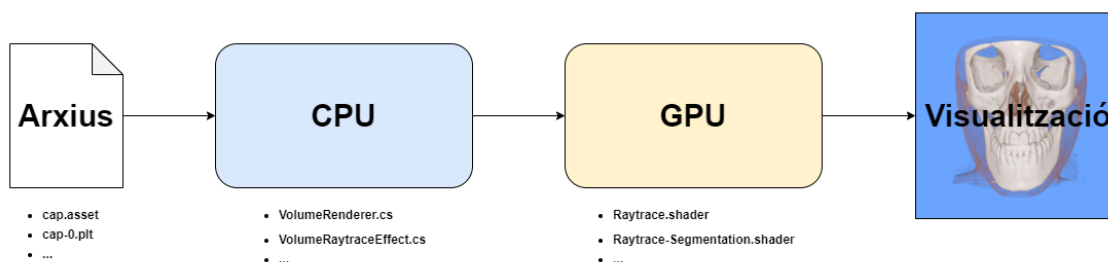


Figura 3.1: *Pipeline* simplificat del projecte original on *cap.asset* és el model de volum, *cap-0.plt* és la TF precalculada, *VolumeRenderer.cs* i *VolumeRaytraceEffect.cs* són les classes principals i *Raytrace.shader* i *Raytrace-Segmentation.shader* són els *shaders* on es realitza el *ray casting* amb les textures d'intensitat i segmentació respectivament. Elaboració pròpia.

En la GPU, el *shader Raytrace* (el qual defineix els dos tipus de *shaders* programables més habituals, el *Vertex* i el *Fragment*, amb especial èmfasi en el segon) calcula el color i la opacitat finals de cada píxel de la finestra de visualització usant la tècnica de *ray casting* (veure Figura 3.2), el pseudocodi de la qual presentem en l'Algorisme 1. Per cada píxel de la finestra es llença un raig, per cada raig que interseca amb

<sup>1</sup>Programa definit per l'usuari i dissenyat per executar-se en alguna fase d'un processador gràfic. Proporcionen el codi per a determinades etapes programables del *pipeline* de visualització.

<sup>2</sup>Variable global de *shader* declarada amb el qualificatiu d'emmagatzematge "uniforme". Aquests actuen com a paràmetres que l'usuari d'un programa *shader* pot passar a aquest programa. S'anomenen així perquè el seu valor és uniforme entre totes les invocacions.

el model volumètric es realitza un mostreig regular en el qual s'accedeix al valor d'intensitat de la textura que representa el model i amb aquest valor s'accedeix a la textura de la TF, amb el que obtenim el color i la opacitat. Després d'aplicar Phong per calcular la il·luminació aquests valors es sumen als valors acumulats del píxel. Aquest procés es repeteix fins que la opacitat és màxima o el raig travessa tot el model. Arribats a aquest punt, ja podem visualitzar la imatge final per la pantalla del dispositiu.

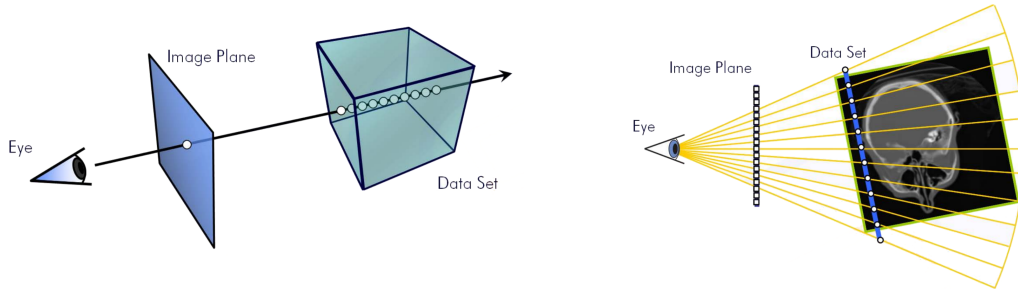


Figura 3.2: Dibuixos il·lustratius de la tècnica de *ray casting* amb models volumètrics on *Image Plane* és la finestra de visualització i *Data Set* és el model volumètric, representat com una textura 3D d'intensitat. Per cada píxel de la finestra és traça un raig que en intersecar amb el model obté el valor d'intensitat cada certa distància. [6]

---

**Algorithm 1:** Volume rendering with ray casting for 1D TF

---

```

Input: rayOrigin, ray, rayLength, sourceDepth, step
Output: (color, opacity)
Data: volume, tf
/* This is for one ray only, used in the Fragment will work for all rays */
1 color = (0.0, 0.0, 0.0);
2 opacity = 0.0;
3 if rayLength != 0.0 then
4     for f = 0.0 to f <= 1.0 do
5         voxelPos = rayOrigin + ray * f;           // Calculate current position
6         voxelDepth = getDepth(voxelPos);
7         if voxelDepth >= sourceDepth then
8             break;                               // Exit if position is beyond volume
9         end
10        /* Get intensity value of current position from texture */
11        voxelValue = volume(voxelPos);
12        /* Get color and opacity values of current intensity from texture */
13        transferMaterial = tf(voxelValue);
14        (vColor, vOpacity) = calculatePhongModel(voxelPos, transferMaterial);
15        color += (1.0 - opacity) * vOpacity * vColor;
16        opacity += (1.0 - opacity) * vOpacity;
17        if opacity >= 1.0 then
18            break;                               // Exit if max opacity
19        end
20        f += step;                               // Advance to the next sampling position
21    end
22    opacity = clamp(opacity, 0, 1);               // 0 <= opacity <= 1
23 end

```

---

Un últim detall que convé mencionar del projecte original és la capacitat de visualitzar models segmentats. En aquest cas, el *pipeline* es manté pràcticament igual però per poder treballar amb el model segmentat primer s'ha d'importar la seva segmentació (mitjançant un complement de Unity) i s'utilitza un *shader* diferent, anomenat *Raytrace-Segmentation*. La part que modifica respecte el que no utilitza segmentació s'il·lustra en el pseudocodi del Algorisme 2, com es pot apreciar s'utilitza una nova textura 3D que representa la segmentació. Aquest *shader* utilitza un color diferent per cada identificador, en total 128 valors compresos

en l'interval  $[0..254]$ , on només els valors parells s'utilitzen per segments, els senars es reserven per les fronteres de cada segment (el que permet tenir més llibertat en la visualització). El valor 254 es considera "no segmentació" i en aquest cas el *shader* es comporta com *Raytrace*. Usant tècniques semiautomàtiques es van classificar els vòxels del model volumètric de cap humà en maxil·lar superior, maxil·lar inferior i dents, la visualització d'aquest model segmentat es pot observar en la Figura 3.3. Com s'ha introduït en el capítol 1, per la TF de la Figura 3.3 a seria impossible obtenir la separació de la Figura 3.3 b en aquests components ja que tots comparteixen el mateix rang de valors d'intensitat (corresponen al mateix teixit i per tant han estat capturats igual).

---

**Algorithm 2:** Segmented volume rendering with ray casting for 1D TF
 

---

**Input:** *rayOrigin, ray, rayLength, sourceDepth, step, voxelPos*

**Output:** (*vColor, vOpacity*)

**Data:** *volume, tf, segmentation*

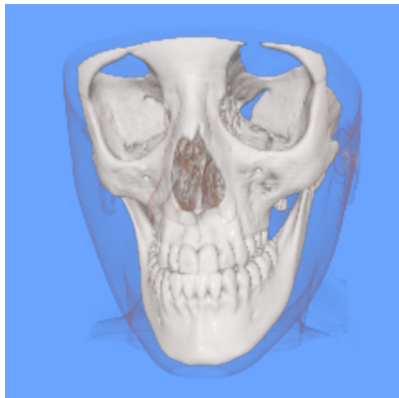
*/\* This code replaces lines 10 to 12 of Algorithm 1 \*/*

```

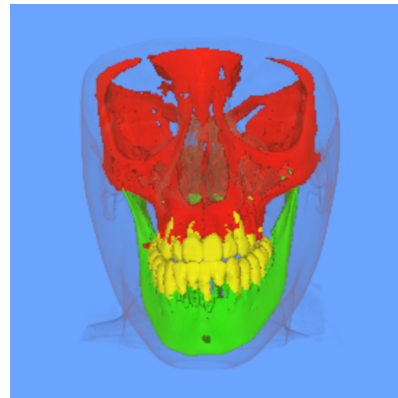
1 voxelValue = volume(voxelPos);
2 segmentationValue = segmentation(voxelPos);           // Get segmentation id
3 transferMaterial = (0.0, 0.0, 0.0, 0.0);
4 if segmentationValue * 256 < 254 then
5     /* Current position has segmentation, get color and opacity of id */
6     transferMaterial = getColorIdSelection(segmentationValue);
7 else
8     /* Current position has no segmentation, do as Algorithm 1 */
9     transferMaterial = tf(voxelValue);
10 end
11 (vColor, vOpacity) = calculatePhongModel(voxelPos, transferMaterial);

```

---



(a) Visualització del model no segmentat amb el *shader Raytrace*



(b) Visualització del model segmentat amb el *shader Raytrace-Segmentation*

Figura 3.3: Comparativa amb el mateix model volumètric de cap humà visualitzant-se amb i sense segmentació. Elaboració pròpia.

## 3.2 Metàfora 1D\_TF

Com ja s'ha definit en el capítol anterior, aquesta metàfora consisteix en el desenvolupament d'una interfície d'edició i de modificació de TF en 2D tradicional adaptada a un entorn VR immersiu. Per tal de poder entendre el que s'ha implementat, primerament, es definiran les funcions de transferència unidimensionals en el marc teòric i, posteriorment, s'exposaran el disseny, la implementació i la interacció.

### 3.2.1 Funcions de transferència unidimensionals

Les funcions de transferència unidimensionals són un tipus de TF, introduïdes en la secció 1.1.1, les quals es caracteritzen per tenir un únic paràmetre d'entrada. Aquestes funcions ens permeten definir característiques

per valors concrets, o rangs, d'intensitat del model volumètric (els valors d'intensitat dels models acostumen a estar normalitzats en l'interval  $[0..1]$ ). Les característiques més habituals per la visualització de models volumètrics són el color i la opacitat, representats amb RGBA (per simplicitat suposarem que també estan en l'interval  $[0..1]$  però si usem 32 bits el més normal es que estiguin en l'interval  $[0..255]$ ). Resulta fàcil veure que la representació gràfica és complexa i difícil d'interpretar, la gràfica resultant és la combinació del eix d'entrada amb els eixos de sortida, és a dir, l'eix d'intensitats amb els eixos R, G, B i A. Per tal de poder simplificar això, assignem el color per separat, i d'aquesta manera podem representar la TF en una gràfica 2D que ens relaciona la intensitat amb la opacitat on cada rang (conjunt de valors consecutius d'intensitat) definit té un RGB concret. Un exemple d'aquest tipus de gràfica es pot veure en la Figura 3.4 a.

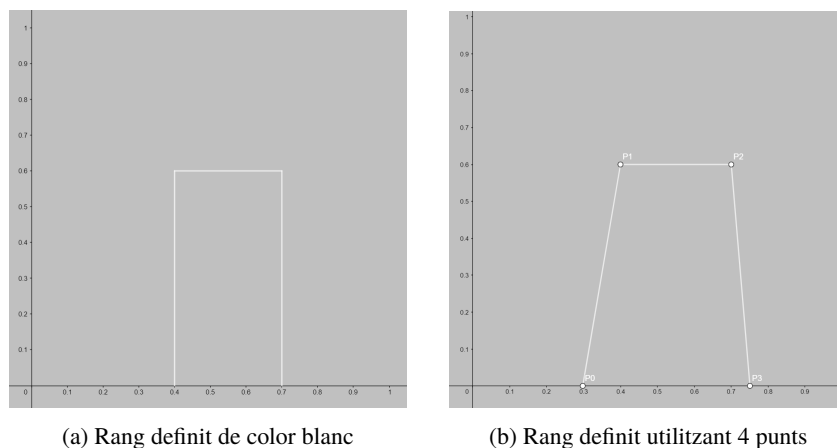


Figura 3.4: Gràfiques que relacionen intensitat (normalitzada, eix abscissa) i opacitat (normalitzada, eix ordenada) amb un rang definit. Elaboració pròpia.

La gràfica anterior, encara que correcte, no és representativa de les que definiríem en un entorn real. Qualsevol persona que hagi treballat amb segmentacions (per exemple acolorit d'imatges) coneix els problemes derivats de la frontera (la part compresa entre dos zones visualment diferenciables o distingibles), quelcom similar succeeix amb els models volumètrics. Suposem que tenim el model d'una mà i volem visualitzar només la part òssia per detectar una possible fractura, el rang concret de valors d'intensitat que representen l'ós ens és desconegut i, potser, ni existeix (degut a la granularitat del mostreig, imprecisions del aparell de captació de dades, etc.). Per minimitzar aquest problema, els rangs es defineixen amb 4 punts (P0, P1, P2 i P3), el que permet transicions suaus en els límits dels cossos que volem destacar, la opacitat incrementa linealment entre P0 i P1, decreix linealment entre P2 i P3 i és manté entre P1 i P2. El resultat final són rangs amb forma de trapezoide com el de la Figura 3.4 b.

Les funcions de transferència unidimensionals són les TF més senzilles, amb només un paràmetre d'entrada, el qual relacionem a una o més característiques de sortida. Existeixen altres més complexes que prenen múltiples paràmetres d'entrada, les quals no tractarem en aquest projecte.

### 3.2.2 Disseny i implementació

Es va decidir construir la metàfora sobre el sistema d'edició de TF unidimensional del que disposa el programa proporcionat (veure Figura 3.5), el qual emmagatzema en la variable *ranges* de la classe *TransferFunction* tots els rangs que formen la TF que després utilitza per construir la textura que s'envia al *shader*. Per tant, es pot modificar la TF si editem aquesta variable i, posteriorment, indiquem a la classe que torni a construir la textura.

Donat que el programa no tenia la capacitat de modificar la TF interactivament en temps d'execució des de l'entorn VR, es va decidir desenvolupar una interfície que ho permetés. Això també em va permetre guanyar familiaritat amb el projecte, ja que aquesta interfície requeria comunicar-se amb les classes principals per obtenir i modificar dades (variable *ranges*). A més, encara que havia treballat amb Unity prèviament, mai en un projecte d'aquesta dimensió, també vaig aprofitar per aprendre a usar alguns element d'aquest motor que desconeixia, com són la creació d'interfícies usant *canvas*, el funcionament de *signals* i *listeners* per elements de UI, la creació de textures per codi o la instanciació d'objectes en temps d'execució.

La interfície es va desenvolupar tenint en compte les particularitats de les UI en VR, per aquesta raó la interfície és un element més de l'escena que l'usuari pot col·locar on desitgi, té els elements interactuables de mida gran (significativament més grans que els de una UI normal) i la precisió dels paràmetres s'ha

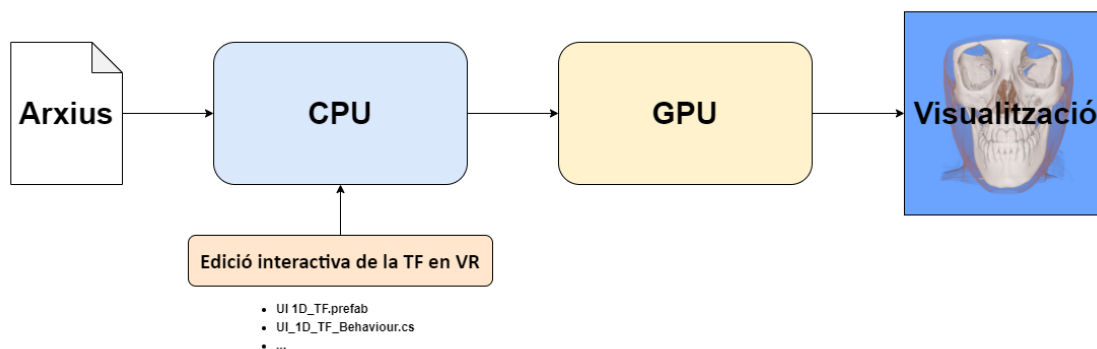


Figura 3.5: Pipeline simplificada de la metàfora 1D\_TF on *UI\_1D\_TF.prefab* és la interfície desenvolupada i *UI\_1D\_TF\_Behaviour.cs* és la classe principal de control de la interfície. Elaboració pròpia.

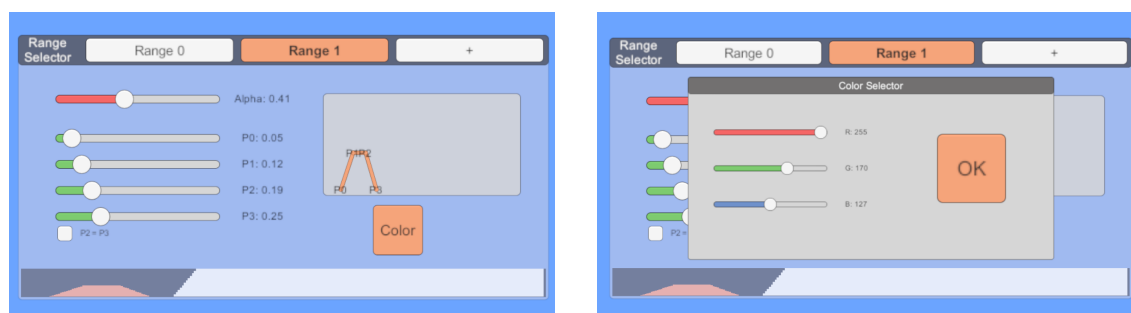


Figura 3.6: Interfície amb nom *UI\_1D\_TF* creada a Unity per editar la TF interactivament. A la part superior trobem la barra de selecció de rangs (en aquest cas hi han 2 rangs definits), el botó '+' serveix per afegir un rang nou. Al centre, d'esquerra a dreta, trobem els sliders i el toggle per definir els valors del rang actual (el rang 1), una gràfica per visualitzar el rang actual i el botó 'Color', amb el color del rang actual, que permet obrir la UI *Color Selector* de modificació del color. A baix, hi ha una textura il·lustrant tota la TF que s'actualitza a cada modificació. La interfície emergent *Color Selector* mostra els 3 sliders per definir el color (R, G i B) i el botó 'OK' per aplicar-lo al rang actual. Elaboració pròpia.

reduït a dos decimals (utilitzar la precisió completa d'un *float* no té sentit quan l'usuari no serà capaç d'aprofitar-ho). La Figura 3.6 mostra el resultat final de la interfície. Aquesta permet seleccionar entre els diferents rangs definits o afegir-ne de nous, definir els valors (P0, P1, P2, P3) del trapezoide de cada rang i el color, visualitzar el rang que s'està editant i veure la funció de transferència en forma de textura.

Per gestionar la interfície s'han creat diverses classes en C#. La classe *UI\_1D\_TF\_Behaviour* n'és la principal, s'encarrega d'inicialitzar la resta i es comunica amb la classe *VolumeRenderer* (conté una instància de la classe *TransferFunction*) per consultar, modificar i afegir els diversos rangs que conformen la funció de transferència. La classe *TF\_Selector\_Behaviour* gestiona la selecció del rang i/o l'addició d'un rang nou. La classe *TF\_Draw\_Behaviour* s'ocupa de la representació gràfica del rang que s'està editant. Finalment, la classe *TF\_Color\_Behaviour* opera la selecció del color del rang seleccionat mitjançant la interfície de la Figura 3.6, inicialment es volia usar un panell amb HUE però la seva complexa implementació i poca precisió van fer que s'optés per 3 sliders que representen l'RGB.

### 3.2.3 Interacció

Pel que fa a la interacció, en aquesta metàfora, és bastant senzilla. En iniciar l'aplicació, l'usuari es troba en un espai virtual delimitat en el qual pot visualitzar el model volumètric seleccionat, aquest s'està dibuixant seguint la TF unidimensional definida a priori. Mitjançant l'activació de l'eina "Edició TF" (el prototip fa servir la tecla F1 però en l'aplicació final això s'hauria de substituir per un element gràfic integrat en l'entorn de VR) l'usuari fa aparèixer o desaparèixer la interfície desenvolupada, amb l'objectiu de dotar de coherència el resultat final aquesta eina sempre fa aparèixer i desaparèixer interfícies, depenent del context (això és, si s'estan utilitzant TF unidimensionals o *style transfer functions*) n'apareixen unes o altres. La UI apareix a una certa distància del usuari, centrada al punter que surt del comandament dret. Aquest mateix



punter és el que serveix per interactuar amb la interfície, ja sigui apuntant a algun dels elements i prement el gallet per tal de modificar la TF unidimensional (els canvis es poden observar en el model al mateix temps) o mantenint premut el botó del sistema per poder canviar-la de lloc. Cada cop que es reinicia l'aplicació els paràmetres tornen als seus valors predefinits. La interacció d'aquesta metàfora es pot veure resumida en el diagrama de la Figura 3.7 amb un cas particular d'ús visible en la Figura 3.8.

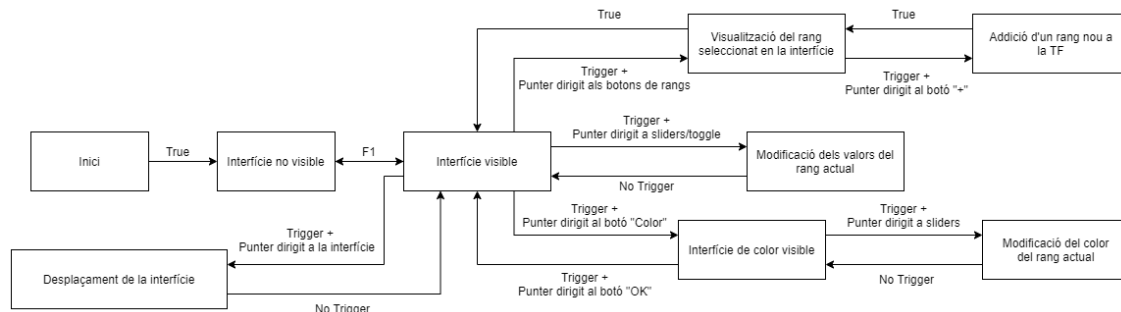


Figura 3.7: Diagrama d'interacció de la metàfora 1D\_TF on la interfície visible és la de la Figura 3.6 i la interfície de color visible és *Color Selector*, la qual es pot veure en la mateixa figura. Com s'especifica, l'usuari pot alternar entre els diferents rangs definits amb els botons superiors o afegir-ne nous amb el botó '+'. També, pel rang actual, pot modificar els 4 punts (P0, P1, P2 i P3) i la opacitat (alpha) que formen el trapezi del rang amb els *sliders*, en els casos en que interressi que els valors de P2 i P3 siguin el mateix es pot usar el *toggle*. Amb el botó 'Color' apareix la interfície *Color Selector* en la qual l'usuari pot definir un nou color pel rang actual. En qualsevol moment es pot moure de lloc la UI per facilitar el treball. Elaboració pròpia.

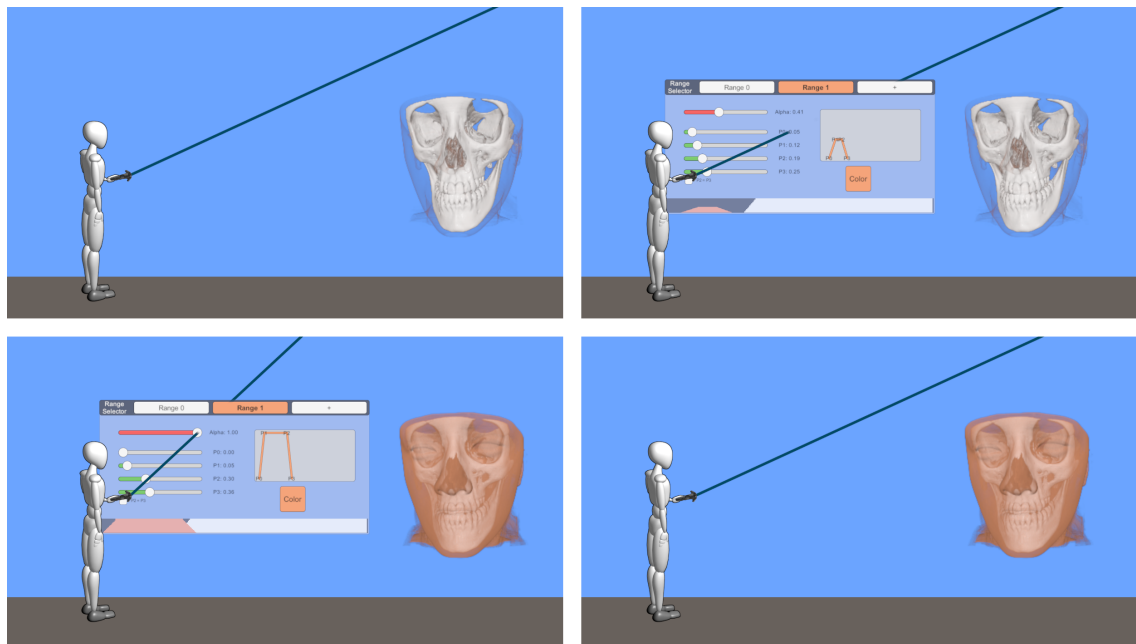
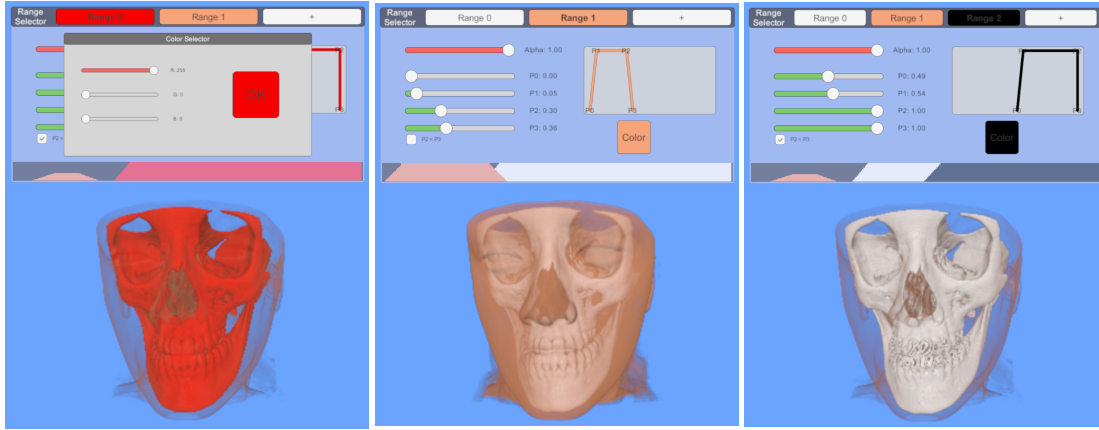


Figura 3.8: Passos a realitzar per modificar la TF amb la metàfora 1D\_TF. Inicialment, només visualitzem el model, després fem aparèixer la UI on assenyalam el punter i modifiquem el valor desitjat de la TF (en aquest cas la opacitat del segon rang). Finalment, fem desaparèixer la interfície. Elaboració pròpia.

La Figura 3.9 mostra el resultat de jugar amb la TF utilitzant la interfície i les interaccions de la metàfora 1D\_TF. Partint del model volumètric de cap humà i la TF predefinida del projecte original (veure Figura 3.1), obtenim una visualització on el crani és de color vermell (Figura 3.9 a) ja que s'han modificat els valors RGB del primer rang, una altra on s'ha incrementat l'opacitat del segon rang al màxim i no s'observa l'ós (la pell ja no és transparent) i l'última (Figura 3.9 c), la qual mostra una part del ós de color negre degut a l'addició d'un nou rang amb aquest color.





(a) Modificació del color del primer rang  
(b) Modificació de la opacitat del segon rang  
(c) Addició d'un rang nou a la TF amb color negre

Figura 3.9: Diferents visualitzacions del mateix model definides usant la interacció proposada i partint de la TF predefinida del projecte original. Elaboració pròpia.

### 3.3 Metàfora Style\_TF

Aquesta metàfora ha consistit en la implementació de les *style transfer functions* en el projecte original amb un disseny adaptat que faciliti l'ús en VR. És el cos del projecte i, per aquesta raó, el seu desenvolupament està més ampliat que el de la 1D\_TF. Anàlogament a com s'ha fet amb l'anterior metàfora, s'explicaran les *style transfer functions* a nivell teòric, a més de, el disseny, la implementació i les maneres d'interaccionar de la Style\_TF.

#### 3.3.1 Style transfer functions

Les *style transfer functions*, d'ara en endavant STF, són un tipus de TF que permeten combinar multitud d'estils d'ombreig diferents en un mateix renderitzat. L'ús de les STF per a la visualització il·lustrativa de models volumètrics, la qual consisteix en destacar característiques importants o eliminar detalls no desitjats mitjançant tècniques de visualització no fotorealista, és ideal ja que aquestes permeten una il·luminació flexible usant dades molt més complexes que la simple assignació de colors i opacitats de les TF unidimensionals.

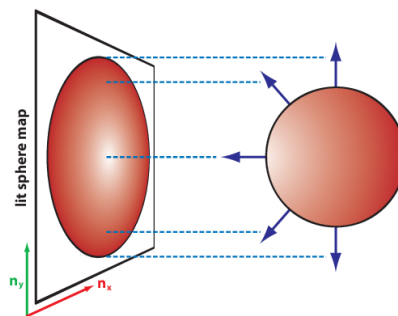


Figura 3.10: Procés d'obtenció del mapa de les *lit spheres*. [17]

En la base de les STF trobem les *lit spheres*, on cada estil es representa amb un mapa<sup>3</sup> d'aquestes. L'obtenció d'aquestes textures s'explica a [24], bàsicament, s'utilitza la projecció ortogonal d'una esfera a la qual s'ha aplicat l'efecte desitjat per obtenir la imatge amb tots els valors d'un dels seus hemisferis. Les *lit spheres* no poden capturar aspectes complexos de la il·luminació realista però sí poden representar l'estil d'ombreig general d'un objecte. A més, la seva dependència de la càmera produeix l'efecte de tenir el focus de llum de l'escena penjant de la càmera, tal com es vol en aquest context.

<sup>3</sup>Una textura quadrada en la qual mai s'accedeix als *texels* exteriors a un cercle inscrit. Per indexar-la s'utilitzen les dues primeres components de la normal en *eye-space*,  $n_x$  i  $n_y$ , després de convertir-les a coordenades de textura, de  $[-1..1]$  a  $[0..1]$ . La component  $n_z$  és ignorada i, per tant, la il·luminació no distingeix entre cares anteriors i posteriors.

Per cada estil també podem definir un contorn, a diferència de les TF unidimensionals. El contorn es visualitza com un cercol a la part més exterior de la circumferència del mapa, el gruix i color d'aquest defineixen el contorn que es visualitzarà. A més, utilitzem les normals de la iso-superfície per minimitzar imprecisions. La Figura 3.11 demostra les bondats d'aquesta aproximació utilitzant un estil verdós amb contorn negre en el crani del model volumètric de cap humà.

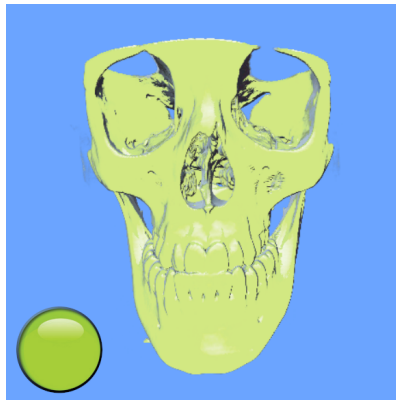


Figura 3.11: Resultat d'aplicar un estil amb contorn a un model volumètric. Elaboració pròpia.

Finalment, per gestionar la opacitat amb les STF tenim dos paràmetres diferents. El primer, anomenat opacitat uniforme ( $a_u$ ), és independent del estil (equivalent a la opacitat de la TF unidimensional). El segon, la opacitat direccional ( $a_d$ ), es defineix en la component *alpha* del mapa de la *lit sphere* i depèn del estil i de la normal del punt. La opacitat final es descriu com  $a = a_u * 0.5 + a_d * 0.5$  en el nostre cas (en [17] s'utilitza  $a = a_d * a_u^{0.5+term}$ , el que permet alguns efectes avançats).

En el que respecta al seu ús, i partint d'un projecte que ja visualitza models volumètrics amb TF regular, [17] descriu una manera d'afegir la STF al esmentat projecte utilitzant tres textures (que es poden interpretar com matrius de dades), les quals presentem a continuació:

- Transfer function texture (*tft*): textura 1D que emmagatzema els valors de  $a_u$  i  $i$  (si el valor és fraccionat vol dir que hi ha interpolació d'estils) per cada valor d'intensitat.
- Index function texture (*ift*): textura 1D que tradueix els valors  $i$  a els  $j$ , necessària degut a que un estil pot ser usat multitud de vegades per diferents valors de rang.
- Style function texture (*sft*): textura 3D formada per un conjunt de textures 2D, les quals són els diferents estils de que disposem.

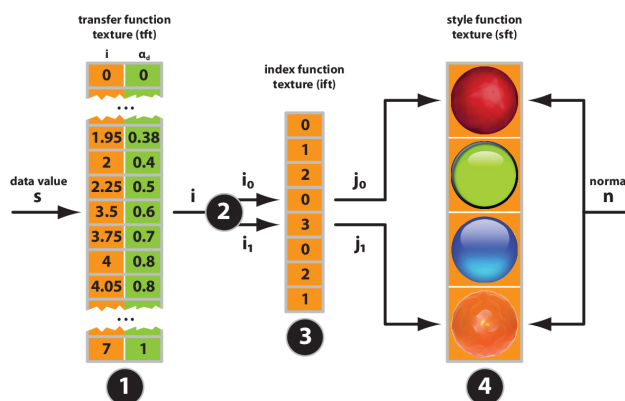


Figura 3.12: Pipeline per obtenir el color de la *lit sphere* a partir del valor d'intensitat del vòxel usant 3 textures. [17]

Per acabar d'entendre com es relacionen aquestes tres textures i com obtenim el color i la opacitat finals a partir d'una densitat d'entrada, tenim la Figura 3.12 i l'Algorisme 3. La primera mostra una possible configuració de les tres textures amb els seus respectius accessos per un valor d'entrada  $s$ . El segon especifica l'ordre d'accessos d'una manera més formal i introdueix la combinació de dos estils.

**Algorithm 3:** *Style transfer function lookup***Input:**  $s, n$ **Output:**  $a_u, color, a_d$ **Data:**  $tft, ift, sft$ 

```

1  $(i, a_u) = tft(s);$ 
2  $i_0 = \lfloor i \rfloor;$ 
3  $i_1 = i_0 + 1;$ 
4  $w = i - i_0;$ 
5  $j_0 = ift(i_0);$ 
6  $j_1 = ift(i_1);$ 
7  $(color, a_d) = sft(n_x, n_y, j_0) * (1 - w) + sft(n_x, n_y, j_1) * w;$ 

```

**3.3.2 Disseny i implementació**

Per a una millor comprensió, la implementació de la metàfora s'ha dividit en els blocs següents, ordenats cronològicament segons es van completar: classe de la STF, *shader* de la STF, modificació interactiva de la STF, model segmentat i classe per a les interaccions. Cal tenir present que aquesta metàfora ha tingut dues fases, una primera (veure Figura 3.13) destinada íntegrament a adaptar el projecte donat per permetre la visualització de models volumètrics amb intensitat utilitzant la STF i, una segona (veure Figura 3.14), en la qual s'ha expandit aquesta capacitat als models segmentats amb noves interaccions.

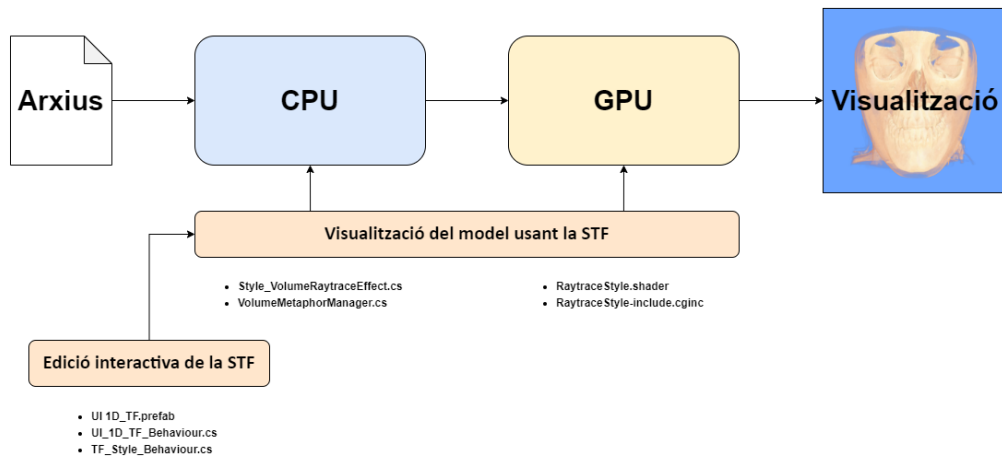


Figura 3.13: *Pipeline* simplificada de la primera fase de la metàfora Style\_TF on *Style\_VolumeRaytraceEffect* és la nova classe encarregada de la STF i *RaytraceStyle.shader* el nou *shader* associat. Elaboració pròpia.

**Classe de la STF**

El primer que es va desenvolupar de la metàfora Style\_TF va ser la classe *Style\_VolumeRaytraceEffect*, la Figura 3.15 mostra el seu disseny mitjançant un diagrama UML. Aquesta classe heretà de *VolumeRaytraceEffect* i s'encarrega de generar les textures necessàries per l'ús de la STF i de passar-les com a *uniforms* al *shader* associat a aquesta metàfora (veure el subapartat següent).

Per utilitzar la STF en el nostre projecte s'han adaptat els passos descrits a [17]. Aquesta proposta permet una relativa fàcil implementació de la STF en un projecte que ja utilitzi una TF, com és el nostre cas. Com s'ha especificat, el projecte original consulta la TF en el *shader* mitjançant una textura. En el nostre, quan s'utilitzi la STF es substituirà aquesta textura per les tres noves: la *tft*, la *ift* i la *sft*.

Una de les principals complicacions del projecte va aparèixer durant el desenvolupament d'aquesta classe, en concret, amb la creació de les textures prèviament esmentades. Per alguna raó, l'accés als valors RGB (les textures segueixen el format RGBA32) al *shader*, retornava valors diferents als definits en la classe. Després d'intentar entendre on estava l'error i provar diverses modificacions, cap exitosa, es va optar per usar només la component *alpha* de les textures (al ser un prototip, es va considerar acceptable). Encara que no es va esbrinar l'explicació a aquest problema, el fet que només afecti als valors RGB sembla indicar que es un problema amb el filtrat de textura.

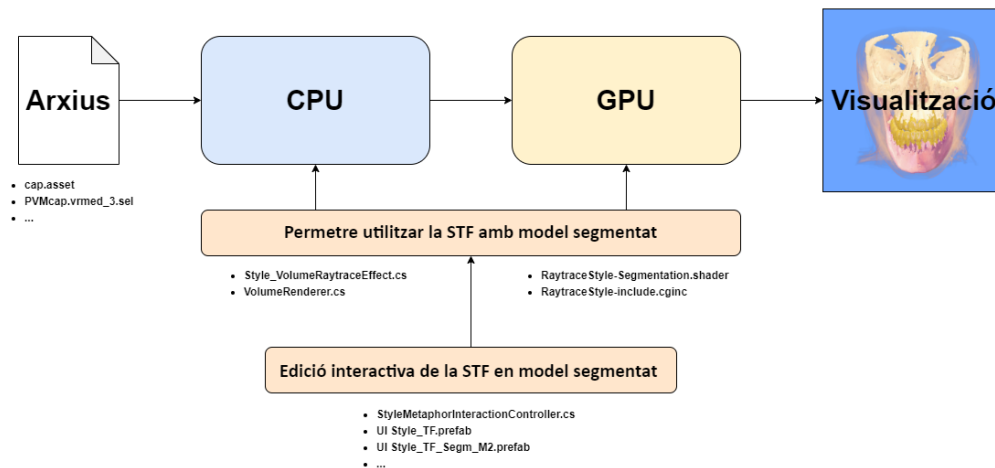


Figura 3.14: Pipeline simplificat de la segona fase de la metàfora Style\_TF on *RaytraceStyle-Segmentation.shader* és el shader que permet usar la STF amb models segmentats i *StyleMetaphorInteractionController.cs* és la nova classe encarregada de gestionar les interaccions. Elaboració pròpia.

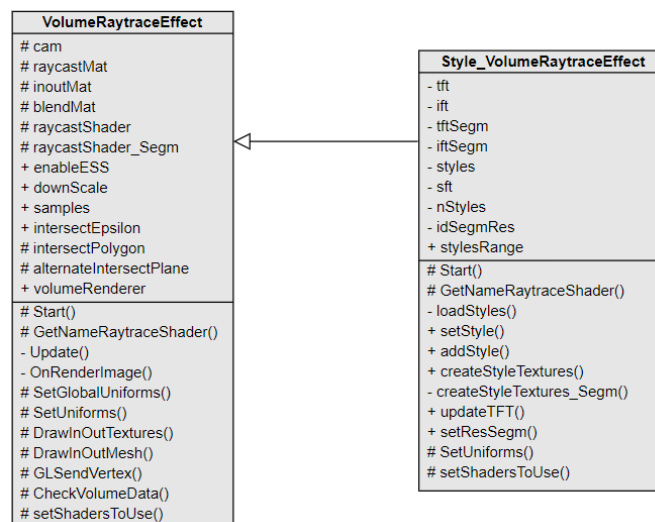


Figura 3.15: Diagrama UML de la classe *Style\_VolumeRaytraceEffect*. Elaboració pròpia.

### Shader de la STF

Paral·lelament a la creació de la classe, també es va desenvolupar el shader *RaytraceStyle*. Basat en *Raytrace* però amb els accessos a textura i el càlcul del color final diferents. S'utilitzen les tres textures creades a la classe *Style\_VolumeRaytraceEffect* per obtenir el color i la opacitat finals, l'accés a aquestes es fa seguint el procediment descrit en l'Algorisme 3. A diferència del projecte donat, no s'aplica cap tècnica d'il·luminació ja que aquesta ve donada pel propi estil.

El pseudocodi del Algorisme 4 mostra la modificació realitzada al shader original per funcionar amb la STF. En concret, s'han reemplaçat les línies 11 i 12 del original per les que es mostren en el nou. Amb el valor d'intensitat i la normal obtenim el color i la opacitat de cada mostreig.

Tornant al exemple de les textures i els seus accessos de la Figura 3.12, amb la variable *voxelValue* accedim a la textura *tft* i obtenim l'índex per accedir a la textura *ift* i una de les opacitats del vòxel. Com aquesta implementació permet la mescla de dos estils, l'índex pot tenir valors decimals (per exemple, si l'índex és 3.4 obtenim els subíndexs 3 i 4 amb contribucions al estil final de 0.6 i 0.4 respectivament). La textura *ift* només es necessària si es vol permetre la mescla entre dos estils per un valor d'entrada, si no, es podria accedir directament amb l'índex de la *tft* a la textura *sft*. La *sft* conté tots els estils disponibles, amb els valors de la *ift* accedim als estils concrets i amb el valor de la normal obtenim el color concret de la *lit sphere* i l'altra opacitat, com s'ha descrit anteriorment. Pel que fa al color, no s'utilitza Phong com a

*Raytrace*, cada estil té la seva pròpia il·luminació (per exemple, la *lit sphere* de color verd amb un cercol exterior negre produeix l'efecte de tenir un cos verd amb contorn negre).

---

**Algorithm 4:** Volume rendering with ray casting for STF
 

---

**Input:** *voxelPos*, *voxelValue*

**Output:** (*vColor*, *vOpacity*)

**Data:** *tft*, *ift*, *sft*

```

/* This code replaces lines 11 and 12 of Algorithm 1 */
1 N = getGradient(voxelPos) ; // Get the normal of the current position
2 ; (i, au) = tft(voxelValue) ; // Get 1st opacity
3 ; i0 = [i] ;
4 i1 = i0 + 1;
5 w = i - i0;
6 j0 = ift(i0);
7 j1 = ift(i1);
8 (vColor, ad) = sft(Nx, Ny, j0) * (1 - w) + sft(Nx, Ny, j1) * w ; // Get color and 2nd opacity
9 vOpacity = au * 0.5 + ad * 0.5 ; // Calculate final opacity

```

---

### Edició interactiva de la STF

Amb la classe i el *shader* ja creats, es va passar al disseny d'una interfície que permetés l'edició interactiva de la STF dins l'entorn de VR. Inicialment es va adaptar la que s'usa amb la metàfora 1D\_TF, de manera provisional, per poder començar a provar la classe i el *shader* el més aviat possible, i al final és va decidir deixar aquesta interfície modificada ja que era bastant còmoda d'utilitzar (veure Figura 3.16).

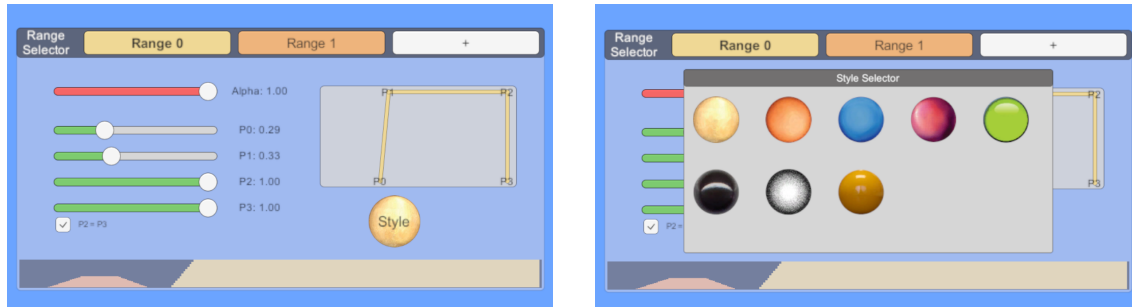


Figura 3.16: Interfície amb nom *UI Style\_TF* creada a Unity per editar la STF interactivament. Elaboració pròpia.

Com es pot apreciar, en comparació amb la UI de la Figura 3.6, tots els apartats relacionats amb el color han estat substituïts per estils. El botó 'Color' ha estat substituït pel botó 'Style', el qual mostra l'estil actual del rang seleccionat i al clicar-lo apareix la interfície emergent *Style Selector*, en comptes de *Color Selector*, amb tots els estils disponibles. No només ha canviat això, en elements on l'ús de la textura no quedava prou bé (com en els botons dels rangs o en la textura de la TF) s'ha optat per usar, per cada rang, un color obtingut de la mitja dels 64 píxels (*kernel* 8x8) centrals de la *lit sphere* seleccionada.

### Model segmentat

Des del principi, un dels objectius de la metàfora *Style\_TF* era explorar maneres d'interaccionar amb models volumètrics segmentats, on l'identificador substitueix la intensitat com a paràmetre d'entrada de la TF. L'ús d'aquest tipus de models ens hauria de permetre interaccions més interessants que les del model basat en intensitats, de fet, l'article en el que ens basem treballa exclusivament amb models segmentats. Gràcies a l'extensió de temps (veure capítol 5) ha estat possible pensar diverses aproximacions, de les quals s'han acabat implementat dos.

Per permetre tot això, s'ha ampliat la classe *Style\_VolumeRaytraceEffect* amb una nova funció per poder crear les 3 textures amb disseny pensat per identificadors. L'accés a les textures segueix sent el de la Figura 3.12 però per accedir a la *tft* ara s'utilitzen identificadors, el que ha obligat a tornar a generar les textures

consultant els identificadors del model segmentat i tenint en compte que els accessos senars són la frontera dels parells. També s'han afegit línies de codi per alternar entre els *shaders*.

Això últim es deu a que s'ha creat un nou *shader*, *RaytraceStyle-Segmentation*, basat en *Raytrace-Segmentation* i *RaytraceStyle* que accedeix a les textures amb la variable *segmentationValue* del vòxel pertinent. De la mateixa manera que en el projecte original, si aquesta variable és 254 el *shader* es comporta com *RaytraceStyle* i, en aquest cas, utilitza les 3 textures del model no segmentat per obtenir el color i la opacitat.

A més, també s'ha creat la capacitat de permetre ressaltar de color groc el segment del model que s'està senyalant amb el punter, el que permet fer saber al usuari quin és el segment que modificarà. Això ha estat possible gràcies a la funció *getCandidateID()* (veure pseudocodi en Algorisme 5) creada a la classe *VolumeRenderer*, la qual utilitza la posició i direcció del punter per calcular els segments amb els que interseca aquest. El identificador del segment es enviat mitjançant el *uniform resSegm* al *shader*, el qual s'ha ampliat per permetre pintar de color groc i amb il·luminació Phong els vòxels amb aquest valor (veure Algorisme 6). Es pot observar el resultat en la Figura 3.17.

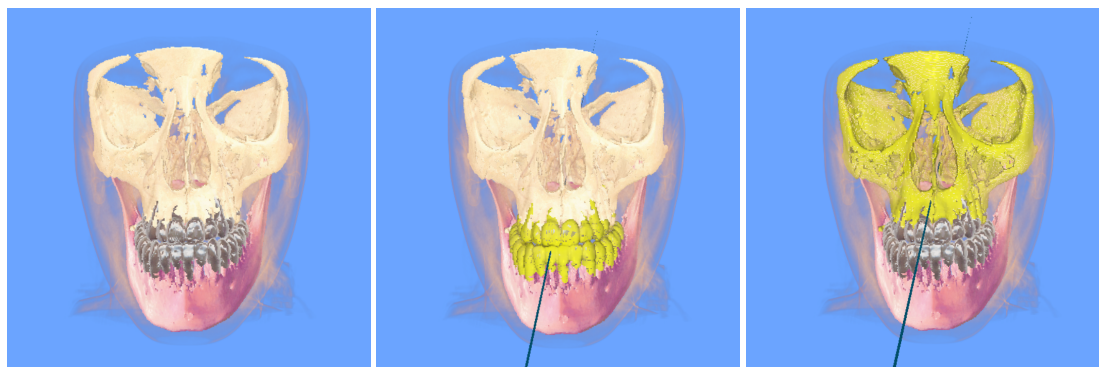
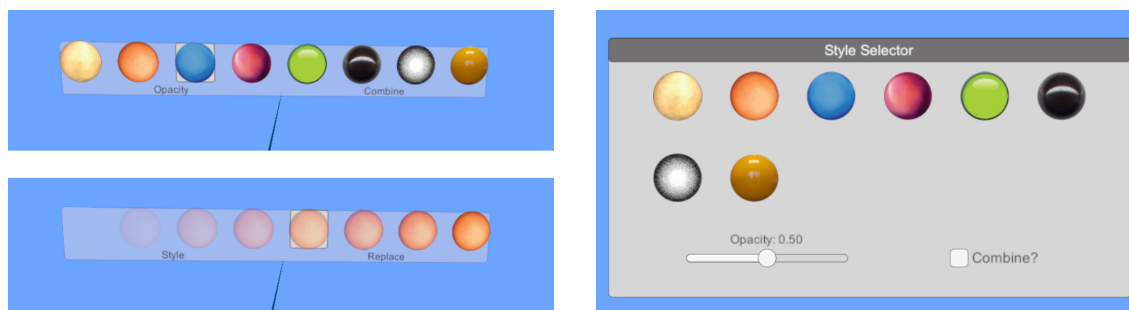


Figura 3.17: Demostració de la capacitat de ressaltar de color groc segments assenyalats amb el punter del model volumètric segmentat de cap humà. Elaboració pròpia.



(a) Interfície amb nom *UI Style\_TF\_Segm\_M1* de la primera aproximació

(b) Interfície amb nom *UI Style\_TF\_Segm\_M2* de la segona aproximació

Figura 3.18: Interfícies de les dues aproximacions diferents a la interacció amb el model segmentat. Elaboració pròpia.

Com ja s'ha comentat, s'han creat dues aproximacions diferents, la primera té una interfície que penja del punter i presenta valors discrets d'estils i opacitats i la segona presenta una interfície estacionària (les dues es poden veure en la Figura 3.18). Es creu que la primera pot ser més adequada per persones habituades als entorns virtuals immersius ja que permet fer accions més ràpidament un cop es domina, la segona presenta una interacció més senzilla i és ideal per a gent amb menys experiència.

### Classe per a les interaccions

L'augment de complexitat en les interaccions amb el model segmentat ens ha portat a crear la classe *StyleMetaphorInteractionController* (veure Figura 3.19) que hereta de *MetaphorInteractionController* i, com el seu nom indica, serveix per gestionar les interaccions. Per exemple, el gatell del comandament en la primera aproximació pot servir tant per aplicar un estil a un identificador (si el punter està en contacte amb

**Algorithm 5:** Get segmentation identifier

---

**Input:** *ray, depth, delta*  
**Output:** *id*  
**Data:** *volume, bounds, segmentation*

```

1 id = 254;                                // No segmentation by default
2 idSet =  $\emptyset$ ;
3 currentDepth = -1;
4 (distance, p, exit) = getIniPos(ray) ;    // Get 1st position
5 if not exit then
6   while p in bounds do
7     currentID = SampleSegmentation(p) ;    // Get id from current position
8     if currentID < 254 ;                    // Is segmentation?
9     then
10      if idSet  $\cap$  currentID ==  $\emptyset$  ;    // Is new segmentation?
11      then
12        idSet = idSet  $\cup$  currentID;
13        id = currentID ;                    // Update id
14        currentDepth = currentDepth + 1;
15        if currentDepth == depth then
16          break;                            // Exit if new id found at depth
17        ;
18      end
19    end
20  end
21  p = p + ray.direction * delta ;    // Advance to the next sampling position
22 end
23 end

```

---

un segment del volum) com per tancar la interfície de selecció (si aquesta es troba activa). L'ús d'una classe exclusiva per gestionar tot això facilita el treball i s'adequa als sistemes del projecte inicial.

### 3.3.3 Interacció

L'augment de contextos, interfícies i accions d'aquesta segona metàfora ha suposat un augment en la complexitat de la interacció. Com ja s'ha explicat, l'aplicació s'inicialitza visualitzant el model volumètric usant la TF unidimensional, si l'usuari prem F2 el model passa a visualitzar-se utilitzant la STF (aquest botó permet alternar entre les dues tècniques en qualsevol moment). En aquest context, l'usuari pot utilitzar F1 per fer aparèixer la interfície que li permeti modificar la STF, com s'ha descrit aquesta és semblant a la de la 1D\_TF però tota la part referent als colors ha estat substituïda per estils. Com la interacció també és molt semblant, s'ha optat per no crear un diagrama específic d'aquesta ja que seria com el de la Figura 3.7 però substituint totes les ocurrences de "color" per "estil".

El que sí es detallarà amb deteniment, per la seva novetat i perquè han estat una part principal de la Style\_TF, són les interaccions concretes que s'han creat per models volumètrics segmentats. Si prèviament s'ha importat una segmentació pel model que es visualitza, podem alternar el mètode de visualitzat perquè tingui en compte la segmentació amb la tecla S. Existeixen dues aproximacions diferents per interaccionar amb models segmentats, amb F1 l'usuari pot alternar entre elles. La Figura 3.20 mostra el diagrama d'interaccions de les dues aproximacions.

La primera aproximació, vincula les interfícies al punter del usuari, per defecte aquest veu una petita finestra que mostra un estil amb una certa opacitat (veure Figura 3.21 a), si usa el *joystick* apareixerà una altre on podrà triar l'estil, la opacitat i el mode (això és si l'estil seleccionat reemplaça l'estil del segment o es combina amb aquest) d'un conjunt discret (veure Figura 3.21 b), amb els valors triats pot confirmar amb el gallet o esperant uns segons. Per modificar l'estil d'un segment, l'usuari només ha d'apuntar amb el punter, el segment seleccionat apareixerà groc temporalment (veure Figura 3.21 c), i prémer el gallet (si el segment desitjat està oclòs, l'usuari pot usar el botó d'adherència per alternar entre les diverses profunditats). Una seqüència d'imatges il·lustrativa de com aplicar un estil i una opacitat a un segment amb aquesta aproximació es pot veure a la Figura 3.21.



**Algorithm 6:** Segmented volume rendering with ray casting for STF

---

```

Input: rayOrigin, ray, rayLength, sourceDepth, step, voxelPos, voxelValue, resSegm
Output: (vColor, vOpacity)
Data: volume, segmentation, tft, ift, sft, tftSegm, iftSegm
/* This code replaces lines 11 and 12 of Algorithm 1 */
1 N = getGradient(voxelPos);
2 segmentationValue = segmentation(voxelPos); // Get segmentation id
3 vColor = (0.0, 0.0, 0.0);
4 vOpacity = 0.0;
5 if segmentationValue * 256 < 254 then
6   (i, au) := tftSegm(segmentationValue); // Current position has segmentation
7   if segmentationValue * 256 == resSegm then
8     transferMaterial := (1.0, 1.0, 0.0, au); // The segment must be highlighted
9     (vColor, vOpacity) := calculatePhongModel(voxelPos, transferMaterial);
10  else
11    i0 = [i];
12    i1 = i0 + 1;
13    w = i - i0;
14    j0 = iftSegm(i0);
15    j1 = iftSegm(i1);
16    (vColor, ad) = sft(Nx, Ny, j0) * (1 - w) + sft(Nx, Ny, j1) * w;
17    vOpacity = au * 0.5 + ad * 0.5;
18  end
19 else
20   /* Current position has no segmentation, do as Algorithm 4 */
21 end

```

---

La segona aproximació, mostra una interfície estàtica amb els estils disponibles (veure Figura 3.22 b), l'usuari selecciona l'estil amb el punter i, si manté el gallet premut, apareix una esfera que apropa al model per seleccionar el segment que vol modificar (veure Figura 3.22 c). Al igual que amb la primera aproximació, la Figura 3.22 mostra com aplicar un estil i una opacitat a un segment amb aquesta segona.

### 3.4 Anàlisi de rendiment

Com ja s'ha especificat anteriorment, un dels factors que s'ha tingut molt present durant tot el desenvolupament de les metàfores és el del rendiment. Bàsicament, garantir unes 90 imatges per segon la majoria del temps. En aquest sentit, podem afirmar que el programa modificat es comporta pràcticament igual que l'original, el qual té una mitja de 120 imatges per segon però oscil·la bastant, entre 90 i 150 imatges per segon.

Amb models segmentats, en apropar-nos molt al model, podem observar una degradació del rendiment que es situa per sota de les 90 imatges per segon. També convé mencionar que la càrrega de la segmentació produeix una aturada de tota la aplicació durant uns minuts. Això, que possiblement sigui un problema de Unity a l'hora de llegir arxius grans, es podria solucionar fàcilment creant un *asset* del motor que fos la pròpia segmentació.

Finalment, especificar que per aconseguir mantenir el mateix rendiment que en l'aplicació original algunes idees s'han hagut de descartar. Per exemple, en una es proposava la visualització del mateix model volumètric múltiples vegades amb diferents TF i l'usuari seleccionava la que més li agradava (qualcom semblant es fa a [16]) però tenir múltiples visualitzacions hauria degradat el rendiment considerablement i es va descartar.

### 3.5 Resultat final

Si es desitja comprovar en què es tradueixen les classes, *shaders* i interfícies creades, es pot observar la Figura 3.23, aquesta mostra la visualització del mateix model volumètric (un cap humà) usant diferents TF. La imatge a mostra el resultat d'usar la TF unidimensional que utilitza el programa base, la qual presenta 2



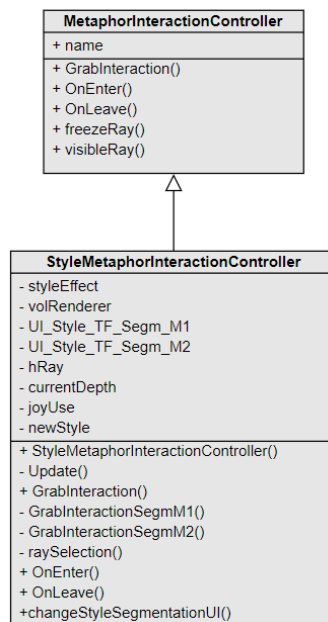


Figura 3.19: Diagrama UML de la classe *StyleMetaphorInteractionController*. Elaboració pròpia.

rangs modificats mitjançant la metàfora 1D\_TF, un color ataronjat per la pell i un grogós pel crani. La b és el resultat d'utilitzar les STF, és pot observar un acabat semblant al de les imatges en llibres mèdics, amb un estil semblant a la pell per aquesta (fins i tot es pot apreciar un efecte de rugositat produït per la pròpia *lit sphere*) i un d'ós per aquest. La imatge c també utilitza les STF però amb el model volumètric segmentat, el que permet coses com ressaltar les dents amb un estil concret (pels vòxels no segmentats s'utilitzen els estils definits per les intensitats, si s'han definit).

A més, en les Figures 3.24, 3.25 i 3.26 es mostren múltiples visualitzacions d'aquest mateix model volumètric utilitzant exclusivament les dues metàfores i partint de la TF original. Com es pot observar, el nombre de visualitzacions diferents és molt gran i encara podria ser-ho més si amplièssim la quantitat d'estils de que disposem (en aquest prototip s'utilitzen 8 estils diferents).

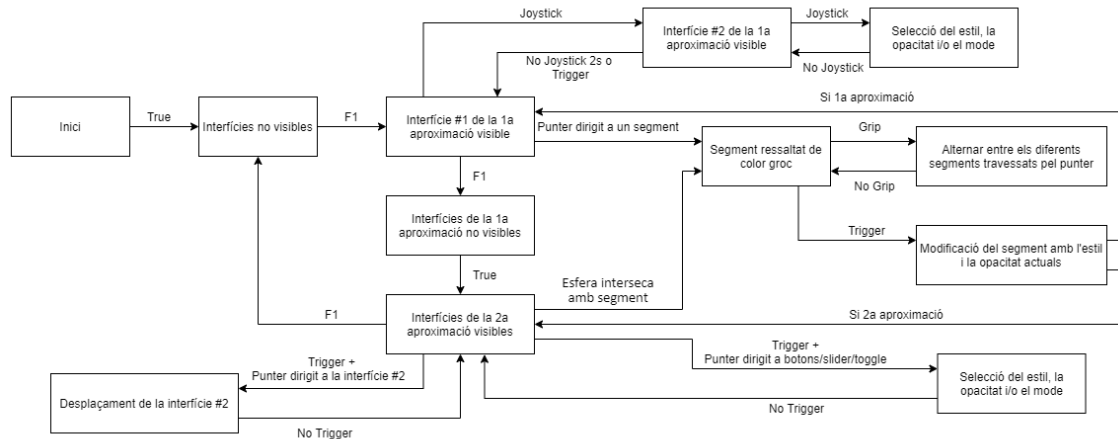
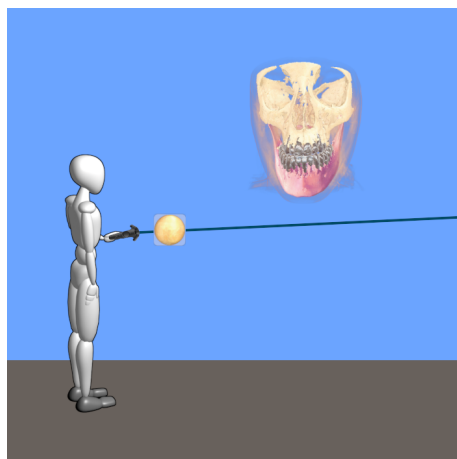
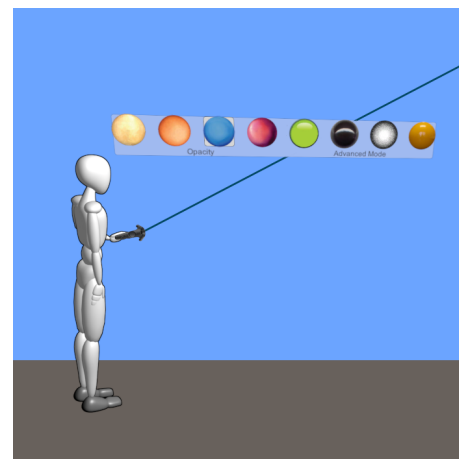


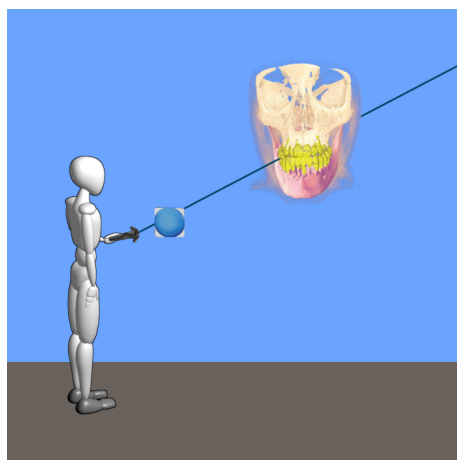
Figura 3.20: Diagrama d'interacció de la metàfora Style\_TF amb models segmentats on les interfícies de la primera i la segona aproximació són les de la Figura 3.18, imatges a i b respectivament. Partim d'un estat inicial en que ja s'ha carregat la segmentació i s'està visualitzant el model segmentat amb la STF. Elaboració pròpia.



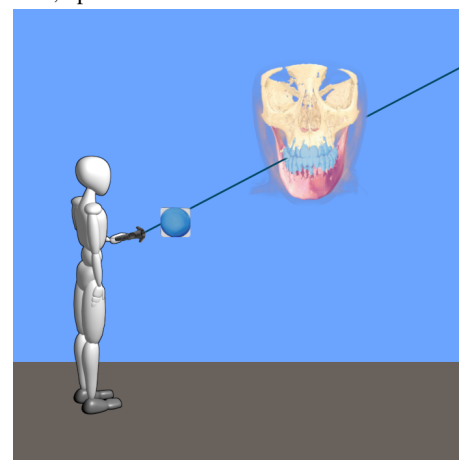
(a) Estat inicial amb estil i opacitat actuals visibles



(b) A la segona interfície es seleccionen els nous estil, opacitat i/o mode

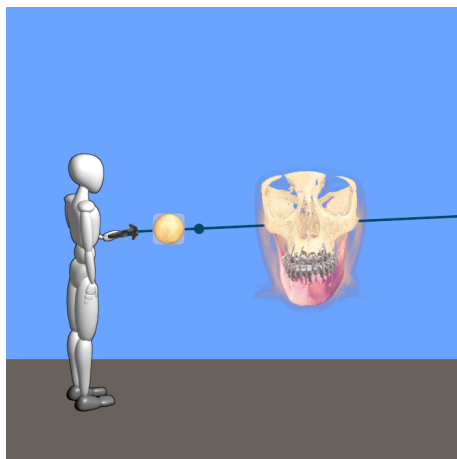


(c) Aplicar els nous estil i opacitat amb el mode seleccionat al segment ressaltat

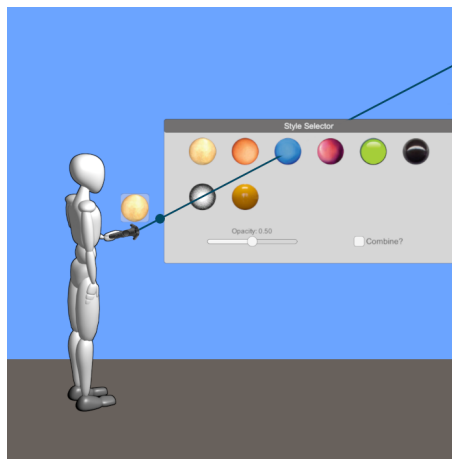


(d) Visualització immediata dels canvis del segment

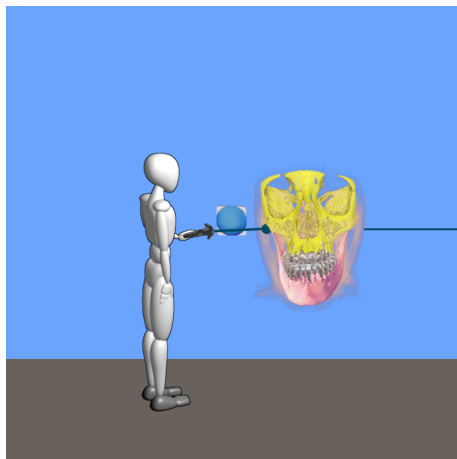
Figura 3.21: Passos a realitzar per modificar l'estil i la opacitat d'un segment amb la primera aproximació. Elaboració pròpia.



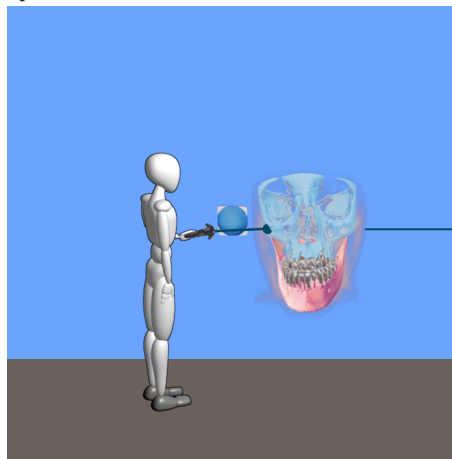
(a) Estat inicial amb estil i opacitat actuals visibles



(b) A la interfície es seleccionen els nous estil, opacitat i/o mode

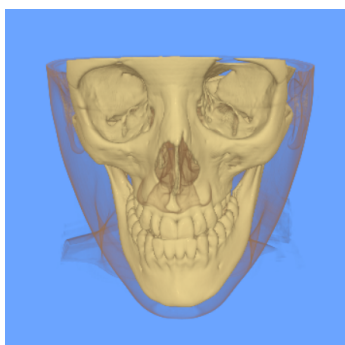


(c) Aplicar els nous estil i opacitat amb el mode seleccionat al segment ressaltat

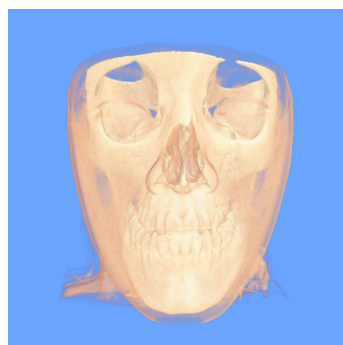


(d) Visualització immediata dels canvis del segment

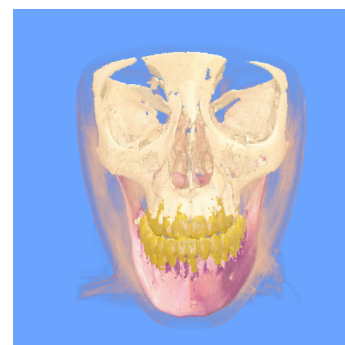
Figura 3.22: Passos a realitzar per modificar l'estil i la opacitat d'un segment amb la segona aproximació. Elaboració pròpia.



(a) Visualització usant la TF unidimensional (metàfora 1D\_TF)



(b) Visualització usant la STF (metàfora Style\_TF)



(c) Visualització usant la STF amb segmentació (metàfora Style\_TF)

Figura 3.23: Visualització del mateix model volumètric usant les diferents tècniques desenvolupades en aquest projecte. Elaboració pròpia.



Figura 3.24: Diferents visualitzacions del mateix model volumètric usant la metàfora 1D\_TF. Elaboració pròpia.



Figura 3.25: Diferents visualitzacions del mateix model volumètric usant la metàfora Style\_TF (sense segmentació). Elaboració pròpia.

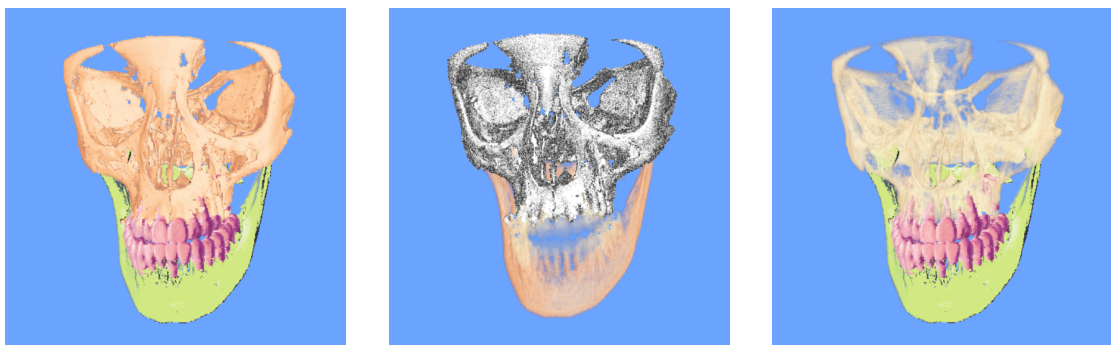


Figura 3.26: Diferents visualitzacions del mateix model volumètric usant la metàfora Style\_TF (amb segmentació). Elaboració pròpia.

## Capítol 4

# Test de validació

El test de validació hauria d'haver consistit en fer provar a un grup significatiu d'usuaris, majoritàriament personal mèdic, el prototip desenvolupat, això és, la versió definitiva de les dues metàfores, per tal d'obtenir dades amb les que poder justificar la veracitat de la nostra hipòtesis en la conclusió. En aquest capítol s'hauria d'haver parlat tant del disseny com dels resultats del test de validació, desafortunadament, la pandèmia de la COVID-19 ha impedit la seva realització. Igualment, s'ha considerat rellevant deixar constància de quina era la intenció amb el test, donat que hauria d'haver estat una part important del projecte.

### 4.1 Disseny del experiment

L'experiment dissenyat buscava generar en l'usuari la necessitat d'usar les dues metàfores. Per aconseguir-ho tots els usuaris partien d'un mateix punt i havien d'aconseguir arribar a un altre, també comú. Per permetre comparacions justes en tots els casos s'utilitza el mateix model volumètric de cap humà (arxiu *cap.asset*) de la Figura 3.23, el qual ens permet importar una segmentació (arxiu *PVMcap.vrmed\_3.sel*) en les proves que ho requereixen. L'objectiu del usuari és aconseguir que el model es visualitzi igual a una referència proporcionada, el que veu al inici és molt diferent de la referència.

En total cada usuari realitzava quatre proves, una dedicada a la metàfora 1D\_TF i tres dedicades a Style\_TF. La primera, consistia en arribar a visualitzar la referència mitjançant la modificació de la TF unidimensional usant les eines creades en la metàfora 1D\_TF. La segona prova consistia en modificar la STF mitjançant la interfície de la metàfora Style\_TF per a models no segmentats. Les dues restants, treballen amb el model segmentat i cada una utilitza una de les aproximacions per a models segmentats desenvolupades en la metàfora Style\_TF. Per tal de reduir el possible condicionament de fer certes proves abans d'altres (per exemple, les dues primeres presenten interfícies molt semblants, la que fem segona resultarà més còmoda ja que estem acostumats de la primera), cada usuari ha realitzat les proves en una assignació aleatòria.

Durant el transcurs del experiment s'haurien enregistrat dades, objectives i subjectives, que després hauríem presentat respectant els criteris estadístics apresos a la carrera. Els paràmetres i qualitats que hauríem tingut en compte, així com la manera d'obtenir-los, es detallen en els apartats següents.

#### 4.1.1 Seguiment del usuari

Es volia poder saber que havia fet cada usuari des de l'inici fins el final de cada prova, la idea era que el recull d'aquestes variables numèriques (dades objectives) permetés reproduir totes les accions que hauria realitzat l'usuari a posteriori per poder-les analitzar i buscar factors comuns. Per tal d'aconseguir-ho es va modificar el projecte per permetre registrar paràmetres concrets a cada fotograma i desar-ho en un arxiu per a la seva posterior consulta. Aquest sistema, que pot ser desactivat mitjançant la variable *track*, degrada el rendiment de l'aplicació en sessions prolongades (l'arxiu cada cop és més gran) però com la durada de cada prova és reduïda, s'ha deixat així (alternativament s'hauria de modificar perquè creï un nou arxiu quan l'actual és molt gran). Les variables numèriques que es van seleccionar són:

- Temps transcorregut: permet mesurar el temps de cada prova, amb el que es pot determinar quina és més ràpida de realitzar.
- Posició i rotació (en graus) del punter: permet mesurar el desplaçament que realitza el braç de l'usuari, amb el que es pot mesurar la fatiga física de cada mètode.

- Posició i rotació (en graus) de la càmera: permet saber on està i cap a on mira l'usuari en tot moment i determinar si es realitzen moviments ràpids que puguin produir fatiga.
- TF utilitzada: indica si s'està usant la TF unidimensional o la STF.
- Segmentació: indica si s'està usant segmentació o no.
- Tecla F1 premuda: indica activació/desactivació d'una interfície per models no segmentats o canvi d'aproximació en models segmentats.
- Tecla F2 premuda: indica canvi de tipus de TF.
- Tecla F4 premuda: indica inici o fi de cada prova.
- Tecla S premuda: permet alternar entre model no segmentat i model segmentat.
- Gallet premut: permet conèixer si s'han hagut de fer moltes interaccions amb les interfícies i segments, un nombre reduït seria l'ideal.
- Botó Grip premut: poques pulsacions d'aquest botó indicaria que l'usuari ha pogut seleccionar el segment oclòs fàcilment.
- Interfície activa: permet conèixer el temps que una interfície d'interacció està activa, amb això es pot determinar si les UI simplifiades per a models segmentats permeten un ús més ràpid.

Amb tot això s'hauria d'haver pogut reproduir el comportament del usuari durant la realització de la prova. A més, s'haurien comparat les tècniques mitjançant les fatigues temporal i física, extrapolades de les variables numèriques recollides, per determinar quina és més còmoda i usable.

#### 4.1.2 Qüestionari

Per obtenir les impressions qualitatives (dades subjectives) dels usuaris es volia elaborar un qüestionari per respondre després d'haver completat les proves. Les preguntes d'aquest qüestionari haurien sigut de resposta tancada (l'usuari tria una nota entre 1 i 5, de molt negatiu a molt positiu) en la seva majoria amb unes poques de resposta oberta. Aquesta aproximació ens hauria de permetre comparar les proves relacionades, és a dir, la primera amb la segona i la tercera amb la quarta. També podríem conèixer si el visualitzat amb STF resulta més entenedor que el de TF unidimensional.

## Capítol 5

# Gestió del projecte

En aquest capítol s'agrupen els diferents apartats relacionats amb la gestió del projecte. En concret, es descriuen la planificació temporal, la gestió econòmica, la sostenibilitat i el compromís social.

### 5.1 Planificació temporal

Aquest TFG ha tingut una durada aproximada de 630 hores, repartides en uns 214 dies, des del 9 de setembre de 2019 fins al 19 d'abril de 2020. Inicialment, es volia defensar el treball al gener però com durant el desenvolupament van aparèixer nous objectius, es va decidir posposar-la al torn d'abril (setmana del 27 d'abril). Aquesta data es va tornar a endarrerir, degut a la pandèmia de la COVID-19, fins el torn de juny (setmana del 29 de juny) per tal d'intentar acabar de perfilar el projecte i realitzar el test de validació un cop reobrisin les instal·lacions universitàries. A part d'això, no s'ha trobat cap altra restricció temporal, donat que l'objectiu d'aquest projecte ha consistit en el desenvolupament d'un prototipus pel grup de recerca ViRVIG i no té cap client final.

De cara al desenvolupament del TFG s'han destinat 3 hores de treball diàries a excepció de dimarts i dijous que s'han realitzat 4, dues d'aquestes al Centre de Realitat Virtual. Cal tenir en compte que certs esdeveniments, com exàmens de la facultat, han fet variar puntualment aquest temps.

#### 5.1.1 Definició de les tasques

A continuació es procedirà a descriure les tasques que venen derivades dels objectius definits en el capítol 2 i conformen aquest TFG. També es justificarà l'estimació temporal assignada a cada una. Per a una major claredat, les tasques es troben emmarcades en grups temàtics però la seva descripció és individual. Podeu trobar un resum de totes elles a la Taula 5.1.

### Gestió del projecte

El concepte gestió de projectes agrupa un conjunt de tasques molt variades però indispensables pel correcte desenvolupament d'un projecte amb una quantitat de feina significativa.

Degut a la metodologia àgil triada, setmanalment s'han realitzat reunions de control amb la codirectora per valorar la feina feta, aclarir dubtes i determinar els objectius dels propers dies. S'ha estimat una durada de 22 hores ja que, generalment, s'han realitzat dues reunions de 30 minuts cada una per setmana, dimarts i dijous. Convé aclarir que prèviament a la data d'inici del TFG es van produir reunions setmanals durant el mes de juliol, de durada variable, les quals també s'han comptabilitzat.

Una part de vital importància a l'inici del treball va consistir en la definició de l'abast del projecte en el context del seu estudi. Es va especificar l'objectiu general del TFG, la contextualització, el perquè de la temàtica (rellevància i justificació), com s'havia de desenvolupar i amb quins mitjans. El temps requerit per completar aquesta tasca van ser 25 hores. Aquesta no es va poder iniciar abans de valorar i triar la millor opció a implementar.

Un cop es va definir l'abast, també va ser important fixar la planificació temporal provisional per a l'execució total del TFG, mitjançant la descripció de les diferents fases del projecte i els recursos i requeriments associats a cada una. El temps requerit per desenvolupar aquesta tasca va ser de 9 hores.

De la mateixa manera, amb l'abast concretat, es va haver de realitzar una estimació del pressupost necessari per la realització del projecte. Aquesta tasca pot ser útil per certs projectes emmarcats en el

Codi	Tasca	Temps	Dependència
<b>T1</b>	<b>1. Gestió del projecte</b>	<b>141 h</b>	
T1.1	1.1 Reunions	22 h	
T1.2	1.2 Abast del projecte	25 h	T2.3
T1.3	1.3 Planificació temporal	9 h	T1.2
T1.4	1.4 Pressupost	5 h	T1.2
T1.5	1.5 Informe de sostenibilitat	5 h	T1.2
T1.6	1.6 Memòria	70 h	
T1.7	1.7 Presentació	5 h	T1.6
<b>T2</b>	<b>2. Treball previ</b>	<b>55 h</b>	
T2.1	2.1 Familiarització amb el programa	25 h	
T2.2	2.2 Cerca d'informació	20 h	
T2.3	2.3 Valoració d'opcions	10 h	T2.2
<b>T3</b>	<b>3. Implementació de la metàfora 1D_TF</b>	<b>125 h</b>	
<b>T3.1</b>	<b>3.1 Primer cicle</b>	<b>85 h</b>	
T3.1.1	3.1.1 Disseny	25 h	T2.3, T1.2
T3.1.2	3.1.2 Implementació	50 h	T3.1.1
T3.1.3	3.1.3 Valoració de possibles millores	10 h	T3.1.2
<b>T3.2</b>	<b>3.2 Segon cicle</b>	<b>40 h</b>	
T3.2.1	3.2.1 Disseny	10 h	T3.1.3, T4.1.3
T3.2.2	3.2.2 Implementació	20 h	T3.2.1
T3.2.3	3.2.3 Valoració de possibles millores	10 h	T3.2.2
<b>T4</b>	<b>4. Implementació de la metàfora Style_TF</b>	<b>240 h</b>	
<b>T4.1</b>	<b>4.1 Primer cicle</b>	<b>150 h</b>	
T4.1.1	4.1.1 Disseny	40 h	T2.3, T3.1.3
T4.1.2	4.1.2 Implementació	100 h	T4.1.1
T4.1.3	4.1.3 Valoració de possibles millores	10 h	T4.1.2
<b>T4.2</b>	<b>4.2 Segon cicle</b>	<b>90 h</b>	
T4.2.1	4.2.1 Disseny	25 h	T4.1.3, T3.2.3
T4.2.2	4.2.2 Implementació	55 h	T4.2.1
T4.2.3	4.2.3 Valoració de possibles millores	10 h	T4.2.3
<b>T5</b>	<b>5. Test de validació</b>	<b>70 h</b>	
T5.1	5.1 Disseny	10 h	T3.2.3, T4.2.3
T5.2	5.2 Implementació	30 h	T5.1
T5.3	5.3 Execució	15 h	T5.2
T5.4	5.4 Valoració de resultats	15 h	T5.3
<b>Total</b>		<b>631 h</b>	

Taula 5.1: Llistat de les tasques del TFG amb el seu codi, les seves dependències i l'estimació temporal. Elaboració pròpia.



context empresarial, però no ho és gaire en el treball actual on només hem volgut crear un prototip per veure si una hipòtesi és correcta. Precisament per això, el temps de dedicació van ser unes 5 hores.

Un altre document que es va haver de generar després de l'abast, però que té poca utilitat en un projecte com aquest va ser el de sostenibilitat, el resultat del informe justifica l'idea. Tenint en compte el que demanava la universitat, es van dedicar 5 hores en la seva elaboració.

També trobem l'elaboració del document principal del TFG, o memòria, format pels documents especificats anteriorment, la documentació referent al programa desenvolupat, els resultats dels tests i la documentació tècnica. El càlcul exacte d'hores no ha estat fàcil ja que és una tasca que s'ha realitzat en paral·lel a les altres activitats al llarg de tot el projecte. Tenint en compte altres treballs escrits del mateix estil fets a la facultat i les actualitzacions del document enregistrades a Overleaf, s'han estimat unes 70 hores de feina.

Finalment, trobem la presentació, aquesta tasca consisteix en la preparació prèvia per a la presentació del TFG, la qual es realitza després d'acabar la memòria. També es prepararà un petit vídeo del programa desenvolupat (inicialment es volia preparar una demostració però la naturalesa telemàtica de la presentació i la falta de material ens ha obligat a canviar-ho). Com en la carrera ja hem practicat diverses vegades les exposicions orals, en principi, només es dedicaran 5 hores. Es requerirà d'un ordinador per les diapositives i la reproducció del vídeo i, si es possible, un petit públic per una representació més realista.

### **Treball previ**

Abans de la data d'inici ja es va començar a treballar en el projecte, la dedicació però va ser més relaxada. Aquí s'agrupen les tasques que es van realitzar durant aquest període.

A partir d'un programa creat additivament entre diversos TFG anteriors es va decidir començar a habitar-se a aquest amb la implementació d'una petita característica que permetés comprendre el seu disseny. La durada va ser de 25 hores. Es van usar els recursos HTC Vive i un ordinador de gamma mitja-alta.

També es va aprofitar l'estiu per cercar informació sobre el renderitzat de volums, els entorns de VR immersiva, les funcions de transferència i altres temes complementaris. Principalment es van consultar llibres i articles científics. Es van destinar 20 hores.

Un cop es van adquirir els coneixements necessaris, es va procedir a pactar una possible millora del projecte. El temps de decisió i debat va durar 10 hores.

### **Implementació de les metàfores**

Aquest grup de tasques en veritat n'agrupa dos, les implementacions de les metàfores 1D\_TF i Style\_TF, però com en general les tasques són similars s'ha decidit ajuntar-los. També cal aclarir que per cada metàfora s'han realitzat dos cicles, per una visió general podeu consultar la Taula 5.1.

La fase de disseny de cada una de les metàfores ha estat de les tasques més importants per tal de minimitzar camins sense sortida en la implementació. Gràcies a la metodologia àgil i la possibilitat de consultar amb experts de la matèria aquesta tasca es va realitzar satisfactòriament. En el primer cicle, el disseny de la metàfora Style\_TF ha estat més llarg degut a la seva complexitat, el temps final ha estat d'un 25 hores en la metàfora 1D\_TF i 40 hores en la Style\_TF. En el segon cicle, bàsicament s'ha millorat el disseny del primer, per això el temps és inferior, unes 10 hores en la 1D\_TF i 25h en la Style\_TF.

La tasca d'implementació ha consistit en traduir els sistemes i gràfics creats en el disseny a codi. Seguint la mateixa explicació d'abans, tenim pel primer cicle unes 50 hores i 100 hores per la 1D\_TF i Style\_TF respectivament que han esdevingut 20 hores i 55 hores en el segon cicle. Com en el treball previ, s'han usat com a recursos les HTC Vive i un ordinador de gamma mitja-alta.

Amb la metàfora ja implementada, s'ha demanat opinió del resultat obtingut a persones properes, membres del grup de recerca, per tal de determinar elements a millorar de cara al segon cicle o a la realització del test d'usabilitat. Després de consultar, sabem que s'han dedicat 10 hores per cada una de les 4 tasques.

### **Test de validació**

Desafortunadament, i encara que es va intentar evitar endarrerint la defensa al juny, no s'ha pogut tenir accés a les instal·lacions ni al equipament necessari per realitzar el test de validació en la seva totalitat. Com hauria estat una part important del projecte s'ha decidit mantenir en la versió final de la memòria per deixar constància de la nostra intenció amb aquest.

Podem desglossar el test de validació, que s'hauria d'haver realitzat amb les dues metàfores, un cop desenvolupades, en les tasques descrites a continuació. Només les dues primeres s'han pogut realitzar.

Primer, es va decidir un model i es van dissenyar mesures de seguiment de les accions del usuari per tal de comprovar si efectivament les *style transfer functions* són millors en un entorn virtual. Amb unes 10 hores va ser suficient.

A continuació, es va haver d'adaptar el disseny triat a les dues metàfores i es van afegir aquells recursos necessaris per a la correcta execució. 30 hores van ser suficients. Els recursos utilitzats haurien d'haver sigut les HTC Vive i un ordinador de gamma mitja-alta, però en aquest punt ja no teníem accés al casc de VR.

Un cop implementat, els participants haurien provat les dues metàfores i contestat una enquesta. Aquesta tasca es podria haver completat en unes 15 hores usant els mateixos recursos que en la implementació.

Finalment, es volia valora el feedback dels participants i determinar quina metàfora és millor, també s'hauria meditat sobre la totalitat del projecte. En aquest punt s'hauria pogut aprofitar tot el temps restant però es van fixar 15 hores.

### 5.1.2 Representació gràfica de la planificació

Complementari a la Taula 5.1 també s'ha usat la tècnica de Gantt per representar la realització de les tasques al llarg del temps. En la Figura 5.1 podeu veure el resultat obtingut amb les dates de inici i finalització de cada tasca i les seves dependències de la planificació inicial, la Figura 5.2 mostra la part de la planificació final (després de canviar el torn de defensa al juny) que ha variat respecte l'inicial.

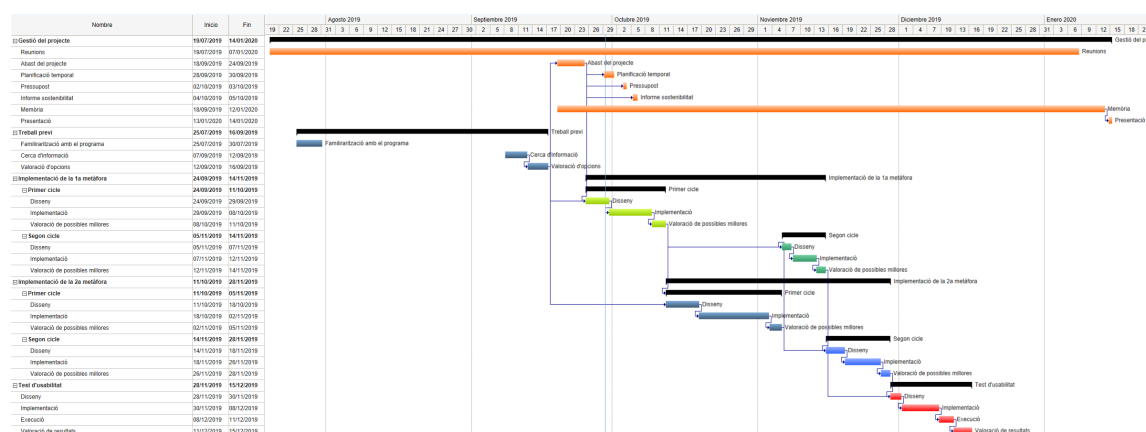


Figura 5.1: Gantt il·lustratiu de la planificació temporal inicial del projecte. Realitzat amb el *plugin* online Gantter. Elaboració pròpia.

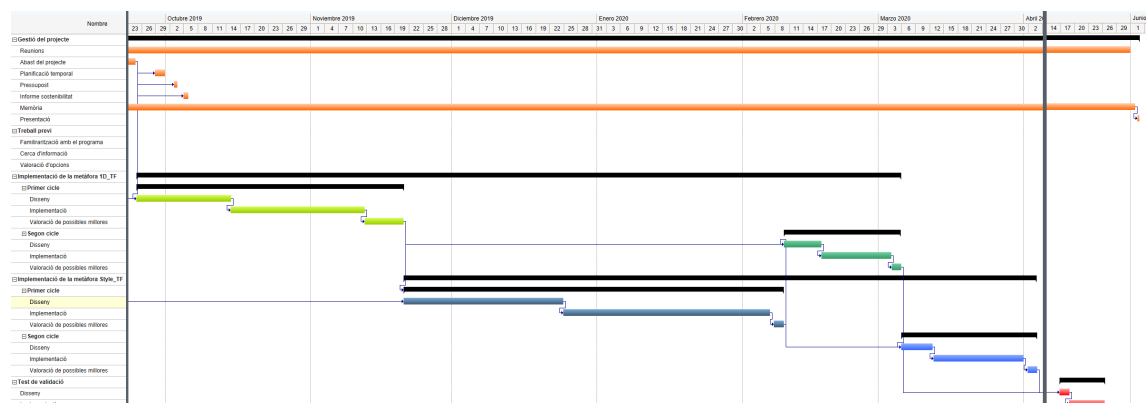


Figura 5.2: Gantt il·lustratiu de la part de la planificació temporal final del projecte que ha variat respecte l'inicial. Realitzat amb el *plugin* online Gantter. Elaboració pròpia.

### 5.1.3 Gestió del risc

En el punt anterior s'ha descrit la planificació temporal que s'ha seguit durant el desenvolupament del TFG. Malgrat tenir una planificació ben definida, en tot projecte de certa envergadura sempre poden, i solen, sorgir complicacions. En aquest apartat es descriuen els principals obstacles i riscos del projecte i els plans alternatius.

#### Implementació de les metàfores

De totes les tasques que s'han definit, les que tenien més possibilitats de desviar-se del temps estimat eren les referents al disseny i implementació de les dues metàfores. Ha resultat especialment complicat predir el temps ja que no s'ha treballat en res similar anteriorment. Degut a la seva major complexitat, també és important destacar que hi havien més possibilitats de desviació temporal en la metàfora Style\_TF.

Una manera de fer front a això ha consistit en afegir dependències al segon cicle de les dues metàfores de manera que aquests no comencin fins que tinguem implementat les primeres versions d'aquestes. Això ha permès refinar les dues metàfores en funció del temps disponible i ens ha garantit que com a mínim tindríem una primera versió operativa de les dues. No ha sigut necessari utilitzar les 25 hores que es van reservar en la planificació inicial.

#### Rendiment

Un dels principals obstacles ha estat el rendiment del programa desenvolupat. Les tècniques de renderitzat de volums, per la seva naturalesa, solen ser molt costoses computacionalment i a això cal afegir el fet que hem treballat en un entorn de VR, el que doble la resolució horitzontal de renderitzat.

Hem sigut molt curosos amb l'eficiència del programa i hem descartat el més aviat possible idees que eren massa exigents en recursos. Les consultes amb la codirectora han d'ajudar a evitar problemes d'aquest àmbit i per aquesta mateixa raó no s'han hagut d'usar les 10 hores reservades en la planificació inicial.

#### Disponibilitat de la VR

En principi, s'ha disposat de les HTC Vive del grup de recerca i això no hauria d'haver canviat de no ser per la pandèmia de la COVID-19.

La solució proposada en la planificació inicial, recórrer a les Oculus Rift d'un conegut que es va oferir a cedir-me les seves, per fer front a la possibilitat de no disposar d'aquest dispositiu durant un temps no ha servit en tots els casos. Encara que aquestes es van utilitzar per avançar feina durant les festes d'hivern, un problema amb el cable de vídeo ha impedit que les hagi pogut utilitzar duran el confinament.

#### COVID-19

És raonable afirmar que ningú espera que una de les pandèmies del seu segle coincideixi amb l'any en que presenta el TFG. Aquest fet no es contemplava en la planificació inicial i, encara que no ha afectat al cos del desenvolupament, els seus efectes s'han fet notar en el test de validació i la polidesa del prototip.

## 5.2 Gestió econòmica

Com a part de la gestió de projectes, en aquest apartat es definiran i analitzaran els diferents costos associats al TFG. Aquests es troben agrupats en les categories de costos de personal per activitat, genèrics, de contingència i d'imprevistos. En els apartats següents es procedeix a desglossar cada un dels conjunts per tal de poder-los justificar.

### 5.2.1 Costos de personal per activitat

Per cada tasca definida en el Gantt (veure Figura 5.2) s'ha estimat un cost, aquest ha estat el cost humà per a la realització de la tasca i, per tan, és una relació directe entre les hores realitzades i el cost per hora del perfil que ha desenvolupat la tasca.

Primerament, es procedirà a concretar un preu per hora pels diversos perfils presents en aquest projecte, els quals són cap del projecte, programador i *tester*. En el cas del cap, ha estat un rol compartit entre el director, la codirectora i jo mateix i pel que fa als *testers* podem diferenciar entre els dels cicles d'implementació de les metàfores (mateixos membres que en cap del projecte) i els del test de validació (personal diferent,

Activitat	Perfil			Cost
	Cap del projecte	Programador	Tester	
<b>1. Gestió del projecte</b>	<b>141 h</b>	<b>0 h</b>	<b>0 h</b>	<b>2820 €</b>
1.1 Reunions	22 h	0 h	0 h	440 €
1.2 Abast del projecte	25 h	0 h	0 h	500 €
1.3 Planificació temporal	9 h	0 h	0 h	180 €
1.4 Pressupost	5 h	0 h	0 h	100 €
1.5 Informe de sostenibilitat	5 h	0 h	0 h	100 €
1.6 Memòria	70 h	0 h	0 h	1400 €
1.7 Presentació	5 h	0 h	0 h	100 €
<b>2. Treball previ</b>	<b>40 h</b>	<b>15 h</b>	<b>0 h</b>	<b>1024 €</b>
2.1 Familiarització amb el programa	10 h	15 h	0 h	425 €
2.2 Cerca d'informació	20 h	0 h	0 h	400 €
2.3 Valoració d'opcions	10 h	0 h	0 h	200 €
<b>3. Implementació de la metàfora 1D_TF</b>	<b>43 h</b>	<b>70 h</b>	<b>12 h</b>	<b>2090 €</b>
<b>3.1 Primer cicle</b>	<b>29 h</b>	<b>50 h</b>	<b>6 h</b>	<b>1420 €</b>
3.1.1 Disseny	25 h	0 h	0 h	500 €
3.1.2 Implementació	0 h	50 h	0 h	750 €
3.1.3 Valoració de possibles millores	4 h	0 h	6 h	170 €
<b>3.2 Segon cicle</b>	<b>14 h</b>	<b>20 h</b>	<b>6 h</b>	<b>670 €</b>
3.2.1 Disseny	10 h	0 h	0 h	200 €
3.2.2 Implementació	0 h	20 h	0 h	300 €
3.2.3 Valoració de possibles millores	4 h	0 h	6 h	170 €
<b>4. Implementació de la metàfora Style_TF</b>	<b>73 h</b>	<b>155 h</b>	<b>12 h</b>	<b>3965 €</b>
<b>4.1 Primer cicle</b>	<b>44 h</b>	<b>100 h</b>	<b>6 h</b>	<b>2470 €</b>
4.1.1 Disseny	40 h	0 h	0 h	800 €
4.1.2 Implementació	0 h	100 h	0 h	1500 €
4.1.3 Valoració de possibles millores	4 h	0 h	6 h	170 €
<b>4.2 Segon cicle</b>	<b>29 h</b>	<b>55 h</b>	<b>6 h</b>	<b>1495 €</b>
4.2.1 Disseny	25 h	0 h	0 h	500 €
4.2.2 Implementació	0 h	55 h	0 h	825 €
4.2.3 Valoració de possibles millores	4 h	0 h	6 h	170 €
<b>5. Test de validació</b>	<b>25 h</b>	<b>30 h</b>	<b>15 h</b>	<b>1175 €</b>
5.1 Disseny	10 h	0 h	0 h	200 €
5.2 Implementació	0 h	30 h	0 h	450 €
5.3 Execució	0 h	0 h	15 h	225 €
5.4 Valoració de resultats	15 h	0 h	0 h	300 €
<b>Total CPA</b>	<b>322 h</b>	<b>270 h</b>	<b>39 h</b>	<b>11074 €</b>

Taula 5.2: Predicció de costos de personal en funció del perfil per les tasques especificades en la planificació temporal. Elaboració pròpia.

major grup). Pel que fa a la programació, ha estat un perfil que he desenvolupat jo en la seva totalitat. En la Taula 5.3 es pot veure el preu per hora final de cada rol, aquest s'ha obtingut a partir dels preus bruts que es defineixen en un estudi de remuneracions de projectes de software de PagePersonnel [25] i tenen en compte la seguretat social.

Perfil	Retribució
Cap del projecte	20 €/h
Programador	15 €/h
Tester	15 €/h

Taula 5.3: Retribucions per cada un dels perfils que participa en el projecte. Elaboració pròpia.

Amb els valors definides anteriorment ja podem procedir a pronosticar els costos associats a cada tasca definida en el Gantt. En la Taula 5.2 s'han dividit les hores totals de cada activitat entre els diversos perfils que participaran en la seva realització i s'ha calculat el preu final de cada una (el que es coneix per cost per activitat o CPA).

### 5.2.2 Costos genèrics

Aquests són costos que es calculen de manera genèrica ja que intervenen en múltiples tasques al llarg del projecte. En el nostre cas són costos genèrics les amortitzacions, l'espai de treball, el consum elèctric i el consum d'internet.

#### Amortitzacions

Les amortitzacions d'aquest TFG són exclusivament de *hardware*. El *software* usat (Unity, SteamVR, Overleaf, MonoDevelop, etc.) és gratuït, inclús Visual Studio que fins fa uns anys era de pagament (encara que s'inclouïa en el paquet de llicències ofert per la UPC) té ara la versió Community, totalment gratuïta.

Pel que fa al *hardware*, les amortitzacions es poden veure a la Taula 5.4, aquestes s'han calculat tenint en compte una durada del projecte de 7 mesos i seguint la fórmula següent:

$$\text{Amortization} = \text{product\_price} * \frac{\text{months\_project}}{\text{months\_product\_lifetime}}$$

Hardware	Preu	Vida útil	Amortització
PC 1 (HTC Vive)	1200 €	84 mesos	100.00 €
PC 2	1600 €	100.8 mesos	111.11 €
HTC Vive	600 €	67.2 mesos	62.50 €
<b>Total</b>			<b>273.61 €</b>

Taula 5.4: Cost dels recursos *hardware*. Elaboració pròpia.

#### Espai de treball

El projecte s'ha desenvolupat al Centre de Realitat Virtual, situat al Campus Sud. D'aquestes instal·lacions, s'ha utilitzat la sala on es troben l'entorn de VR i els ordinadors de desenvolupament. L'espai de treball utilitzat té un cost aproximat de 200 €/mes, és a dir, un cost total de 1.400 € durant els 7 mesos que ha durat el projecte. Donat que aquest espai està compartit amb tres projectes, considerem just repartir aquest cost entre tots els projectes, per tan el total seria de 466.67 €.

#### Consum elèctric

Després de comprovar el cost actual del kWh, el qual és de 0.1198 €/kWh [26], podem estimar el cost de subministrar energia al equipament definit anteriorment a partir d'una aproximació de les hores d'ús. Aclarir que només es comptabilitza quan els aparells estan encesos. En la Taula 5.5 es poden veure els costos individuals i totals derivats del consum energètic.

Hardware	Potència	Hores actiu	Consum	Cost
PC 1 (HTC Vive)	450 W	300 h	135 kWh	16.17 €
PC 2	700 W	700 h	490 kWh	58.70 €
HTC Vive	6 W	150 h	0.9 kWh	0.11 €
<b>Total</b>				<b>74.98 €</b>

Taula 5.5: Cost del consum elèctric del *hardware*. Elaboració pròpia.

### Consum d'internet

La tarifa d'accés a internet és d'uns 60 € al mes. Si tenim en compte que el projecte té una durada de 7 mesos i que la majoria de dies la jornada laboral és de 3h, la part del cost del consum d'internet d'aquest projecte és de  $7 \text{ mesos} * (60 \text{ €} / 1 \text{ mes}) * (3 \text{ h} / 24 \text{ h}) = 52.5 \text{ €}$ .

### 5.2.3 Costos generals del projecte

La Taula 5.6 resumeix els costos generals justificats en les seccions anteriors. Conjuntament amb els costos de personal per activitat conformen els costos totals.

Concepte	Cost
Amortitzacions	273.61 €
Espai de treball	466.67 €
Consum elèctric	74.98 €
Consum d'internet	52.5 €
<b>Total CG</b>	<b>867.76 €</b>

Taula 5.6: Predicció dels costos generals. Elaboració pròpia.

### 5.2.4 Altres costos

#### Contingència

Contemplant la possibilitat de que poguessin sorgir imprevistos durant el projecte, va ser necessari preparar un fons de contingència per tal d'estar preparats per afrontar qualsevol imprevist. Es va optar per ser caut i es va fixar un nivell de contingència del 15%, el que ha suposat uns  $(\text{Total CPA} + \text{Total CG}) * 0.15 = (11074 \text{ €} + 867.76 \text{ €}) * 0.15 = 1791.26 \text{ €}$ .

#### Imprevistos

Finalment, també hem de valorar el cost que ens pot suposar l'aplicació de plans alternatius en cas d'imprevistos al llarg del desenvolupament del projecte. La Taula 5.7 conté el cost final dels plans per solucionar els imprevistos mencionats en apartats anteriors, aquest preu és calcula a partir del preu i la probabilitat de la incidència.

Imprevist	Preu	Risc	Cost
Noves ulleres HTC Vive	600 €	15 %	90 €
Nou PC	1000 €	10 %	100 €
Increment del temps de disseny (10 h)	200 €	10 %	20 €
Increment del temps d'implementació (20 h)	300 €	20 %	60 €
<b>Total</b>			<b>270 €</b>

Taula 5.7: Costos generats per imprevistos. Elaboració pròpia.

### 5.2.5 Costos totals del projecte

En la Taula 5.8 s'exposa el cost total predit pel projecte, calculat a partir del cost global de les categories prèviament esmentades.

Activitat	Cost
Costos totals	11941.76 €
Total CPA	11074 €
Total CG	867.76 €
Contingència	1791.26 €
Imprevistos	270 €
<b>Total</b>	<b>14003.02 €</b>

Taula 5.8: Predicció dels costos totals del projecte. Elaboració pròpia.

### 5.2.6 Control de gestió

Durant el transcurs del projecte s'ha realitzat un seguiment exhaustiu de les hores empleades en cada activitat per tal d'assegurar que la desviació dels costos predits sigui nul·la o mínima. S'ha usat com a mètrica el desviament del cost per tarifa, relació del cost real amb l'estimat ((cost estimat – cost real) \* hores) i el desviament de l'eficiència amb ((consum hores estimat – consum hores real) \* cost). També s'ha procurat evitar l'ús de recursos no descrits que tinguin substituïts esmentats en aquest apartat (com el rebuig del paper i la utilització de documents digitals). En el nostre cas no es necessita fer la gestió de cap matèria primera ni de ventes ja que no formen part d'aquest projecte. Tampoc ha calgut millorar l'eficiència de les tasques ja que no s'ha detectat un augment del cost real respecte al predit.

## 5.3 Sostenibilitat i compromís social

### 5.3.1 Autoavaluació

L'enquesta dissenyada per EDINSOST ha servit per comprovar la meua actitud respecte la sostenibilitat en les tecnologies de la informació i la comunicació (TIC). Després de realitzar-la sóc conscient de que dispo de un nivell mitjà-alt de coneixements de sostenibilitat en aquesta agrupació. D'això n'he pres consciència amb l'enquesta, ja que bastantes preguntes estan relacionades amb el nivell de coneixements d'àrees concretes de sostenibilitat de les TIC.

També m'agradaria poder dir que m'implico per assegurar que els projectes TIC en que participo tenen ben present la sostenibilitat però el cert és que encara no he participat en cap projecte d'aquest caire. Sí que puc afirmar però que en el meu cas tendeixo a prioritzar la sostenibilitat a l'estalvi econòmic, de manera que si hi ha una alternativa sostenible tendiré a defensar-la malgrat sigui més costosa.

Aquesta enquesta m'ha ajudat a veure que hi ha aspectes de la sostenibilitat aplicada a les TIC que encara desconec però m'agradaria criticar el fet que el tema no guarda relació amb la meua especialitat i que, malgrat haver especificat no tenir experiència laboral, no s'ha tingut en compte que no he treballat anteriorment a l'hora de formular les frases.

### 5.3.2 Dimensió ambiental

L'impacte ambiental de la realització del projecte ha recaigut, principalment, en dos factors: la producció de l'electricitat requerida pel desenvolupament del prototip i els components que conformen el *hardware* que s'ha utilitzat. Existeixen altres factors, com l'impacte de mantenir el servidor on s'allotja el nostre projecte o el poc paper utilitzat per revisar la memòria, però els considerem insignificants en comparació als dos primers. En la secció anterior s'ha comptabilitzat el consum elèctric del *hardware* utilitzat (Figura 5.5), no s'ha pogut reduir l'impacte donat que el consum elèctric ha estat el mínim necessari per poder desenvolupar el prototip. De la mateixa manera, tampoc s'ha pogut minimitzar més, a part del fet que s'han reaprofitat per a un altre projecte, l'impacte ambiental dels components dels dispositius utilitzats. Dubto que pogués reduir els recursos utilitzats si tornés a fer el projecte, en tot cas al anar més ràpid gracies a l'experiència obtinguda podria ser més ràpid i reduir lleugerament el consum elèctric.

En principi, la vida útil del projecte acaba després de reflexionar sobre les dades obtingudes en el test de validació. Si es volgués estendre (el prototip és perfectament usable), es requeriria d'un ordinador de classe mitja-alta i d'unes ulleres de VR immersiva. L'impacte ambiental d'aquests objectes serà el de la producció d'electricitat necessària pel seu funcionament i, en cas de fallida, és molt important portar-los a un punt de reciclatge per minimitzar l'impacte. En general, el projecte empitjora la petjada ecològica però ho fa molt poc i a canvi permet millorar la qualitat de treball d'una àrea tan crítica com la medicina.

L'únic escenari que pot fer augmentar la petjada ecològica és si un dels dispositius *hardware* s'espatlla. Això implicaria fer-se càrrec del dispositiu que no funciona (de la manera com s'ha indicat anteriorment) i buscar un substitut que minimitzés l'augment de la petjada ecològica com, per exemple, un dispositiu de segona mà o un que mostrés d'on provenen els seus materials.

### 5.3.3 Dimensió econòmica

En l'apartat de gestió econòmica d'aquest mateix capítol s'ha quantificat el cost de la realització del projecte. Durant el transcurs del TFG s'ha anat comptabilitzant la despesa, tal i com és descriu en el control de gestió, per garantir que es respectessin els pressupostos fixats. Només ha variat el cost un cop es va ampliar el temps de realització, però això ha estat degut a una decisió meditada i no pas un mal càlcul de previsió. Personalment, crec que aquesta part del informe de sostenibilitat no té gaire sentit aplicada a aquest projecte donat que no es realitza per a una empresa ni ha suposat una gran inversió econòmica.

Donat que com a prototip no té sentit, ni era la intenció, buscar benefici econòmic, el cost durant la vida útil serà el cost dels dispositius necessaris per fer-lo funcionar, en aquest cas, un bon ordinador i les HTC Vive. Aquest *hardware* acostuma a tenir una llarga vida útil (més de 5 anys) però alguns elements, com la goma de subjecció de les ulleres, poden necessitar substitucions més freqüents. En els primers 5 anys segurament podríem passar amb uns 50 €, més enllà, hauríem de substituir els dispositius per noves versions.

L'únic escenari que pot perjudicar la viabilitat del projecte és aquell en el que el centre mèdic no disposa del pressupost necessari per reparar un dels dispositius prèviament esmentats i no pot utilitzar el prototip desenvolupat. És una situació inusual ja que el cost relatiu no és tan elevat si parlem d'un hospital.

### 5.3.4 Dimensió social

L'elaboració d'aquest projecte m'ha fet reflexionar bastant però també és cert que moltes coses han passat en el que portem de 2020. Una de les reflexions en les que he pensat ha estat en com podem aprofitar la tecnologia per ajudar-nos a tots, a la humanitat en el seu conjunt i al planeta, i com de difícil pot resultar fer-ho quan ets en una empresa, on uns nombres abstractes tenen més valor que la moral més bàsica. Una altra ha estat a nivell personal ja que considero que haver fet un projecte d'aquesta magnitud m'ha ajudat a estar més preparat professionalment, durant el grau he vist molts temes però en la majoria de casos no s'observava la connexió entre ells, en el TFG sí.

Com ja s'ha dit prèviament, els beneficiaris d'aquest projecte seran especialistes mèdics que podran explorar els models volumètrics dels seus pacients per tal de practicar operacions o diagnosticar malalties. Aquests especialistes però poden veure's en la situació de patir mareigs durant l'exploració en VR, una petita part de la població pateix aquest fenomen. Tampoc podrien usar aquest sistema persones amb problemes visuals aguts i per persones que utilitzin ulleres un ús prolongat pot arribar a ser molest per la cara. També, per evitar la transmissió de la COVID-19 s'hauria de desinfectar el dispositiu de VR (visor i comandaments) cada cop que un nou usuari l'utilitzi (això és si no es pot garantir un sistema de VR per cada usuari, el que tampoc seria complicat tenint en compte que cada cop són més assequibles). A falta de realitzar el test de validació, creiem que el nostre prototip presenta una manera més còmode i entenedora d'interactuar amb la TF i visualitzar el model, que és el que es volia.

No considerem que el projecte pugui ser perjudicial per algun segment particular de la població en cap cas ni tampoc creiem que pugui generar dependència, la visualització interactiva de models volumètrics en medicina ha de ser complementària a les tècniques tradicionals.



## Capítol 6

# Conclusió

En aquest TFG s'han desenvolupat la “metàfora d'interacció amb funció de transferència unidimensional” (metàfora ID\_TF) i la “metàfora d'interacció amb *style transfer function*” (metàfora Style\_TF) per permetre la visualització interactiva de models volumètrics en un entorn de VR immersiu. La primera, més senzilla, ha esdevingut una interfície que representa una adaptació força literal de la manera en que s'editen les TF unidimensionals en un monitor. La segona, molt més complexa, parteix de la base de la primera però s'enriqueix amb la claredat dels estils de la STF, l'ús de models segmentats i la simplificació de les interfícies, el que permet interaccions molt més satisfactòries i ràpides.

Malauradament, i encara que es va intentar evitar endarrerint la presentació del projecte al estiu, la veracitat de la hipòtesis plantejada no ha pogut ser provada amb dades, ni objectives ni subjectives. La pandèmia de la COVID-19 ens ha impedit realitzar el test de validació amb professionals mèdics i tampoc ens ha permès acabar de comprovar l'òptim funcionament de les interaccions en VR perquè no s'ha tingut accés a les HTC Vive del CRV durant les fases finals del desenvolupament.

Per treballs futurs, el primer que s'hauria de fer és el test de validació per tal de poder acceptar o refutar la hipòtesis plantejada. Amb les dades obtingudes del test es podria elaborar un text científic amb les nostres conclusions que podria ajudar a la comunitat de recerca d'aquesta àrea. Respecte al projecte, podríem intentar anar més lluny ampliant la interacció de l'usuari, en la metàfora Style\_TF, permetent-li classificar regions del model de volum mitjançant la manipulació interactiva (actualment es classifiquen a priori). També, seria interessant adaptar-lo per poder ser utilitzat en ambients diferents al mèdic com, per exemple, un museu on el visitant pot explorar un entorn juràssic i seleccionar entre un conjunt de models volumètrics de dinosaures que es visualitzen usant estils creats amb l'ajut de paleontòlegs i es permet fer focus a estructures d'interès (pell, òrgans, ossos, etc.).



# Bibliografia

1. DOTEVERYON. *Medium*. “We shape our technology; then technology shapes us.” [online]. 2018 [cons. 2019-09-21]. Disponible a: <https://medium.com/doteveryone/https-medium-com-doteveryoneuk-we-shape-our-technology-then-technology-shapes-us-36d16251176e>.
2. PREIM, Bernhard; BARTZ, Dirk. *Visualization in medicine : theory, algorithms, and applications*. 1a ed. Burlington: Morgan Kaufmann, 2007. ISBN 9780123705969.
3. SJSCHEN. *Wikipedia*. File:CTSkullImage.png - Wikimedia Commons [online]. 2005 [cons. 2019-10-09]. Disponible a: <https://commons.wikimedia.org/wiki/File:CTSkullImage.png>.
4. DAVEYNIN. *Wikipedia*. File:UPMCEast CTscan.jpg - Wikipedia [online]. 2012 [cons. 2019-10-09]. Disponible a: [https://en.wikipedia.org/wiki/File:UPMCEast\\_CTscan.jpg](https://en.wikipedia.org/wiki/File:UPMCEast_CTscan.jpg).
5. GLITZY QUEEN00. *Wikipedia*. File:SADDLE PE.JPG - Wikimedia Commons [online]. 2007 [cons. 2019-10-09]. Disponible a: [https://commons.wikimedia.org/wiki/File:SADDLE\\_PE.JPG](https://commons.wikimedia.org/wiki/File:SADDLE_PE.JPG).
6. ENGEL, Klaus; HADWIGER, Markus; KNISS, Joe; REZK-SALAMA, Christof; WEISKOPF, Daniel. *Real-Time Volume Graphics*. 1a ed. Wellesley: A K Peters, 2006. ISBN 9781439864296.
7. DÍAZ IRIBERRI, Jose. *Enhanced Perception in Volume Visualization*. 2013. Tesi doctoral. Universitat Politècnica de Catalunya.
8. MAN, Avi; HADAD, Yzhak; ILUZE, Moshe; BERLINER, Shlomo; ZIV-BARAN, Tomer; ROZENBAUM, Zach; AVIRAM, Galit. Prevalence of increased ratio between the right and left ventricles among patients undergoing CT pulmonary angiography with and without evidence of pulmonary embolism. A: *European Congress of Radiology 2018*. Viena: European Respiratory Society (ERS), 2018, vol. 52, pàg. 855. Núm. 62. Disponible a DOI: 10.1594/ecr2018/C-1805.
9. LAVIOLA, Joseph J. Jr.; KRUIJFF, Ernst; MCMAHAN, Ryan P.; BOWMAN, Doug A.; POUPYREV, Ivan. *3D User Interfaces: Theory and Practice*. 2a ed. Boston: Addison-Wesley Professional, 2017. ISBN 9780134034324.
10. *ViRVIG* [online] [cons. 2019-10-13]. Disponible a: <https://www.virvig.eu/>.
11. ECKER, Kait. *The Minnesota Daily*. Art department’s VR studio sees new uses at UMN [online]. 2019 [cons. 2019-10-13]. Disponible a: <https://www.mndaily.com/article/2019/01/n-art-departments-vr-studio-sees-new-uses-at-umn>.
12. VIGLIAROLO, Brandon. *TechRepublic*. 10 pieces of military technology that changed the way the civilian world works [online]. 2016 [cons. 2020-05-27]. Disponible a: <https://www.techrepublic.com/pictures/10-pieces-of-military-technology-that-changed-the-way-the-civilian-world-works/>.
13. ANTHONY, Sebastian. *ExtremeTech*. 3D TV is dead [online]. 2013 [cons. 2020-05-27]. Disponible a: <https://www.extremetech.com/extreme/145168-3d-tv-is-dead>.
14. LJUNG, Patric; KRÜGER, Jens; GROLLER, Eduard; HADWIGER, Markus; HANSEN, Charles D.; YNNERMAN, Anders. State of the Art in Transfer Functions for Direct Volume Rendering. *Computer Graphics Forum*. 2016, vol. 35, núm. 3, pàg. 669 - 691. ISSN 14678659. Disponible a DOI: 10.1111/cgf.12934.
15. PINKER, Alexander. *Medialist*. VR Healthcare: Virtual Reality in der Medizin [online]. 2018 [cons. 2019-10-09]. Disponible a: <https://medialist.info/2018/01/26/vr-healthcare-virtual-reality-in-der-medizin/>.
16. JÖNSSON, Daniel; FALK, Martin; YNNERMAN, Anders. Intuitive Exploration of Volumetric Data Using Dynamic Galleries. *IEEE Transactions on Visualization and Computer Graphics*. 2016, vol. 22, núm. 1, pàg. 896 - 905. ISSN 10772626. Disponible a DOI: 10.1109/TVCG.2015.2467294.

17. BRUCKNER, S.; GRÖLLER, M. E. Style transfer functions for illustrative volume rendering. *Computer Graphics Forum*. 2007, vol. 26, núm. 3, pàg. 715 - 724. ISSN 14678659. Disponible a DOI: 10.1111/j.1467-8659.2007.01095.x.
18. Unity. Unity Real-Time Development Platform | 3D, 2D VR & AR Visualizations [online] [cons. 2019-09-23]. Disponible a: <https://unity.com/>.
19. HTC. VIVE™ | VIVE Virtual Reality System [online] [cons. 2019-09-23]. Disponible a: <https://www.vive.com/us/product/vive-virtual-reality-system/>.
20. FONS SÀNCHEZ, Joan. *Inspecció interactiva i immersiva de models volumètrics. Aplicació diagnòstica mèdica*. Barcelona, 2017. Treball de final de grau. Universitat Politècnica de Catalunya.
21. RIES, Eric. *The Lean Startup : How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. 1a ed. New York: Crown Publishing Group, 2011. ISBN 9780307887894.
22. GitLab. The first single application for the entire DevOps lifecycle - GitLab [online] [cons. 2020-02-10]. Disponible a: <https://about.gitlab.com/>.
23. Atlassian. Trello [online] [cons. 2020-02-10]. Disponible a: <https://trello.com/>.
24. SLOAN, Peter-Pike J; MARTIN, William; GOOCH, Amy; GOOCH, Bruce. The Lit Sphere: A Model for Capturing NPR Shading from Art. A: *Proceedings of Graphics Interface 2001*. Ottawa, 2001, pàg. 143 - 150. Disponible a DOI: 10.20380/GI2001.17.
25. PAGEPERSONNEL. *PagePersonnel*. Estudios de remuneración 2016. Tecnología. Selección y trabajo temporal especializado. [online]. 2016 [cons. 2019-10-05]. Disponible a: [www.pagepersonnel.es](http://www.pagepersonnel.es).
26. LOZANO GASCO, Raquel. *Compañías de Luz*. Precio kWh España: Información y tarifas 2019 [online]. 2019 [cons. 2019-10-06]. Disponible a: <https://www.companias-de-luz.com/precio-de-la-luz/kwh/espana/>.